NGA.SP.0009.01_1.0.1_SIFR
2019-08-02

National System for Geospatial Intelligence (NSG) and United States MASINT System (USMS) Sensor Integration Framework (SIF) Standards Profile (SP) Reference View

(2019-08-02)

Version 1.0.1

NATIONAL CENTER FOR GEOSPATIAL INTELLIGENCE STANDARDS

Change Log

Version	Approval Date	POC	Change Description
1.0.0	10/19/17	C. Heazel	GWS/GWG Review Draft
1.0.1	08/02/2019	C. Heazel	Revised references to include ontology, updated enterprise integration.

Forward

Sensing systems come in many shapes and sizes. From complex space-based telescopes which measure the background radiation of the Universe, to disposable stick-on thermometers. Likewise the degree of access to sensing systems varies widely. From direct connections to the high-speed Internet to hanging off the end of a low-speed, low quality, intermittent communications link. Yet, it is highly desirable that any authorized user should have access to any sensor and the data that it produces, from anywhere, and at any time. Clearly there is no single suite of technology which can do that. Likewise there is no single set of standards which can support that goal. This is the problem that the Sensor Integration Framework Standards Profile (SIF-SP) attempts to solve.

The SIF-SP establishes an architecture framework to decompose sensing systems into their constituent parts, and identify standards suitable for each of those parts. This framework is defined at two levels. The Reference View (RV) provides an abstract architecture framework. This level is agnostic to any specific technology. It captures the essence of what a sensor system needs to do regardless of how it is implemented and what domain it targets. Technical Views (TV) apply the Reference View architecture framework to a specific technology environment. TVs not only provide specific instruction on how to implement the SIF using a specific technology, they also specify how that implementation maps back into the Reference View. By tracing every Technical View back to the Reference View, the ability to achieve interoperability across technology environments is greatly enhanced.

This specification is the Reference View of SIF-SP version 1.0.1. It provides the abstract SIF model for implementation by the Technical Views.

Table of Contents

1 General

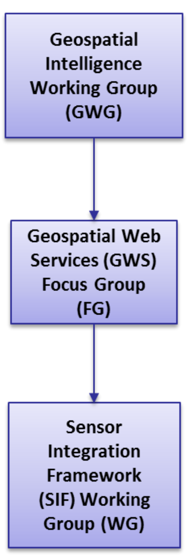
1.1 Background

The purpose of this document is to provide guidance required for sensor data producers and consumers to implement a sensor information enterprise that meets operational requirements, achieves United States (U.S.) Department of Defense (DoD) and Intelligence Community (IC) Chief Information Officer (CIO) goals, and conforms to applicable policy. Additionally, this profile shall define conditions, specifically those applicable to defense computing environments limited by functional mission areas. This profile, while originating from the National Systems for Geo-Spatial Intelligence (NSG) and U.S. MASINT System (USMS) communities, is designed to accommodate the broadest range of sensor information use cases possible. Sensor information implementers can expect this document to 1) identify a collection of necessary standards; 2) constrain those standards to an adequate level of detail; 3) extend those standards as needed; 4) provide overall guidance to employ those standards together.

This version of the Sensor Integration Framework Standards Profile (SIF-SP) has been informed by two existing implementations. MASBUS is a proof of concept developed by the United States MASINT System (USMS). MASBUS Provides an OGC Sensor Observation Service (SOS) integrated with the Distributed Data Framework (DDF) and, by association, the Distributed Common Ground System (DCGS) family of systems. The Integrated Sensor Architecture (ISA) was developed by Night Vision and Electronics Sensors Directorate (NVESD) of the Communications-Electronics Research Development and Engineering Center (CERDEC) of the U.S. Army. It provides an infrastructure and standards that enables data, information, and capability sharing within the tactical environment. Integration of ISA and MASBUS using a preliminary version of the SIF-SP was demonstrated at Enterprise Challenge 2017.

1.2 Scope

This SIF-SP is produced by the Sensor Integration Framework Working Group (SIFWG) of the Geospatial Web Services (GWS) Focus Group (FG) of the Geospatial-Intelligence Standards Working Group (GWG). The GWG serves as a U.S. Department of Defense (DoD), Intelligence Community (IC), Federal, and Civil community-based forum to advocate for IT standards and standardization activities related to GEOINT. The GWG performs two major roles:



1) As a Technical Working Group (TWG) of the DoD and IC CIO Joint Enterprise Standards Committee (JESC); and

2) As a coordinating body for the GEOINT community to address all aspects of GEOINT standards.

The SIF-SP describes an architecture and standards framework for the integration of sensors and sensor systems across all deployment environments. As such, the scope of the SIF-SP is overarching among the community and reaches across multiple areas of interest horizontally rather than vertically. It provides a framework which is applicable to all sensors, regardless of the intelligence discipline. Since almost all sensors have a spatial-temporal component, the GWG was chosen as the most appropriate authority to manage this work.

The purpose of the SIF-SP is to define a framework for the integration of standards-based capabilities. This profile is built around an architecture which is representative of sensing systems and the systems that use them. Standards are then mapped onto that Architecture providing the specifications needed for implementation. The SIF architecture is documented in two levels. This Reference View presents an architecture which is independent of any implementing technology. The concepts presented in this Reference View should be applicable to any implementation environment. Technical Views apply the architecture within the constraints of a specific implementing technology. As there are multiple environments where sensing systems are deployed, so also there are many Technical Views. Each Technical View is scoped to the technology constraints of a specific implementation environment.

1.3 Objectives

The primary objective of the SIF-SP is to further interoperability among DoD and IC sensors and sensor systems. Sensors provide real-time information on the state of the operational environment. This critical information must be accessible where and when it is needed. Too often, sensor information is not accessible due to standards non-compliance, compliance with the wrong standards, or incompatible data and technologies. Some of this is unavoidable. Currently, there are few standards which are appropriate for all possible technological environments. Furthermore, the large number of standards make selection of the correct standard difficult. The SIF-SP strives to address these issues in two ways:

1) Assist in the selection of appropriate standards based on the operational and technical requirements for the planned system,
2) Identify and, where necessary, define the transformation logic needed to bridge one technological environment with another.

1.4 Composition

The SIF-SP is defined in two levels. This Reference View provides a technology agnostic architecture framework for the integration of sensors and sensor systems. Technical Views address the implementation of the Reference View given the constraints of a specific technological environment. There are multiple Technical Views but only one Reference View.

The SIF is augmented with supporting engineering artifacts. Both are accessible on the NGA GitHub site https://github.com/ngageoint/Sensor_Integration_Framework .:

- SIF-SP UML Model: This is an integrated UML model, which includes not only SIF constructs, but also constructs defined by other organizations and activities such as ISO, ISA, NSG, W3C, and ODNI.
- SIF-SP Ontology: This is an OWL ontology of the SIF concepts.

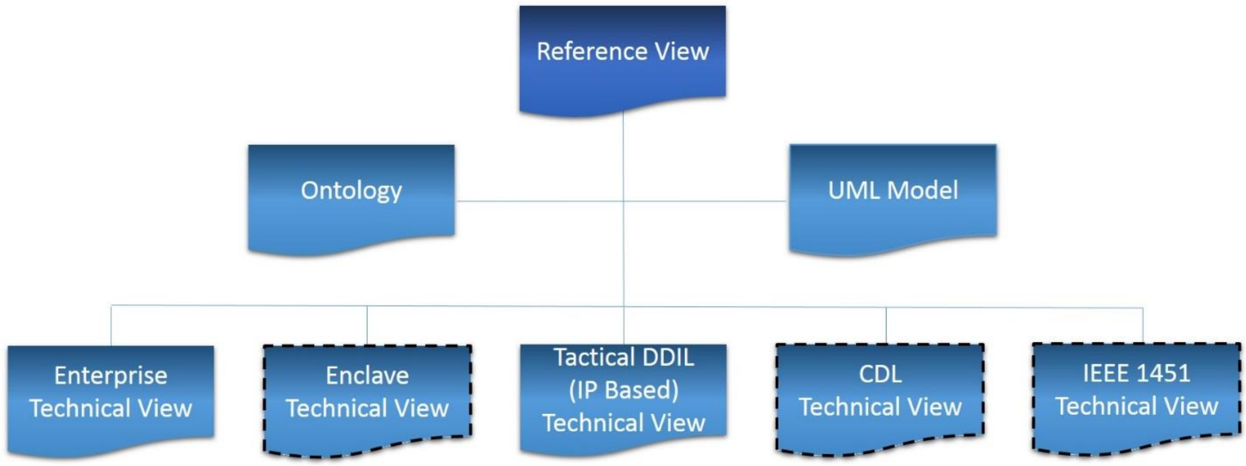


Figure 1: Composition of the Sensor Integration Framework Standards Profile

Five technical environments have been identified to-date. These environments are listed below and discussed in Section 10 (Engineering and Technical Viewpoints). In this release of the SIF-SP, only the Enterprise and Tactical DDIL IP Technical Views are published, TV-1 and TV-3, respectively. The remaining three will be addressed with the next version of the SIF-SP.

1) Enterprise – IP with Web Services and global resources
2) Enclave – IP with Web Services but isolated from global resources
3) Tactical – DDIL IP networks.
4) Common Data Link (CDL) – Airborne non-IP networks
5) IEEE 1451 – Wire protocols

2 Architecture Approach

2.1 Techniques

There are many tools and techniques available to the systems architect. The challenge is to select the set which best addresses the architectural issues under consideration. Four system modeling constructs were leveraged for the Sensor Integration Framework Standards Profile:

- DoD Architecture Framework (DoDAF),[1] contributes an architecture data model to describe the components which comprise the SIF.
- Reference Model of Open Distributed Processing (RM-ODP)[2] provides a set of architectural views to decompose and scope the problem space.
- Unified Modeling Language (UML)[3] object models provide the Object Oriented modeling tools to represent and validate the architecture.

[1] DoD Architecture Framework, http://dodcio.defense.gov/Library/DoD-Architecture-Framework/, Accessed 01 Feb, 2017
[2] From RM-ODP Wiki site at http://www.rm-odp.net/, Accessed on 01 Feb 2017
[3] Object Management Group, *Unified Modeling Language*, version 2.4.1

- Web Ontology Language (OWL) Ontology[4] provides a common vocabulary for use in the SIF-SP documentation and to discuss SIF related issues.

None of these techniques is used in its pure form. They have been freely adapted to address the needs of this profile.

2.2 Architecture Viewpoints

The Architecture used in the Sensor Integration Framework Standards Profile is based on the Reference Model of Open Distributed Processing (RM-ODP)[5]. The RM-ODP is the result of a joint effort by ISO, IEC and ITU-T to provide a coordinating framework for the standardization of open distributed processing (ODP). RM-ODP is also used by the Open Geospatial Consortium (OGC) in the OGC Reference Model (ORM).

RM-ODP defines five Viewpoints. A *Viewpoint* (on a system) is an abstraction that yields a specification of the whole system related to a particular set of concerns. The five Viewpoints defined by RM-ODP have been chosen to be both simple and complete, covering all the domains of architectural design. These five Viewpoints are:

- **Enterprise Viewpoint**, which is concerned with the purpose, scope and policies governing the activities of the specified system within the organization of which it is a part; (See section 0)
- **Information Viewpoint**, which is concerned with the kinds of information handled by the system and constraints on the use and interpretation of that information; (See Section 8)
- **Computational Viewpoint**, which is concerned with the functional decomposition of the system into a set of objects that interact at interfaces - enabling system distribution; (see Section 9)
- **Engineering Viewpoint**, which is concerned with the infrastructure required to support system distribution; (See Section 10 and individual Technical Views)
- **Technology Viewpoint**, which is concerned with the choice of technology to support system distribution. (See Section 10 and individual Technical Views)

The five Viewpoints of RM-ODP support a decomposition of a problem as we move from the Enterprise Viewpoint down to the Technology Viewpoint. The Enterprise Viewpoint deals with the operational requirements. Computational and Information Viewpoints address how the operational requirements can be addressed. Up to this point the architecture is largely conceptual. As such, these three Viewpoints align with the SIF-SP Reference View. The Engineering and Technology Viewpoints address how the Computational and Information Viewpoints can be implemented within specific technology and engineering constraints. These two Viewpoints align with the SIF-SP Technical Views. In practice this alignment is not quite as clean as we would like, but it is sufficient for our purposes. Table 1 describes how each RM-ODP Viewpoint will be used in both the SIF-SP Reference and Technical View documents.

RM-ODP Viewpoint	Reference View	Technical View
Enterprise	Provides a generic description of the capabilities and activities associated with making sensors an enterprise resource.	May provide more detail as to how the Reference View would be implemented given the technology constraints. May be omitted.

[4] W3C Recommendation, *OWL Web Ontology Language*, https://www.w3.org/TR/owl-ref/, Accessed 02 October 2017

[5] From RM-ODP Wiki site at http://www.rm-odp.net/, Accessed on 01 Feb 2017

RM-ODP Viewpoint	Reference View	Technical View
Information	Defines the information concepts required to achieve the operational capabilities described in the Enterprise View. Does not specify formats and encoding.	Refines Reference View information concepts into data formats and encodings. Includes mappings back into the Reference View.
Computational	Identifies the abstract services which are necessary to achieve the operational capabilities described in the Enterprise View. Does not specify specific operations or protocols.	Refines Reference View service concepts into specific service definitions and protocols. Includes mappings back into the Reference View.
Engineering	Specific to an implementation environment. Not covered in the Reference View.	These Viewpoints together describe the technical constraints governing a single Technical View. They address the communications and processing constraints as well as the available support services.
Technology	Specific to an implementation environment. Not covered in the Reference View.	

Table 1 SIF-SP Architecture Views

2.3 DoDAF Architecture Concepts

While the SIF-SP derives its architecture views from RM-ODP, the architecture concepts come from the DoD Architecture Framework (DoDAF) version 2.0[6]. These concepts are defined in the DoDAF Meta-Model (DM2) and the associated UML model. The DoDAF DM2 is a large and complex model. It is far more than what is needed for this profile. Therefore, the SIF will make use of a simplified sub-set of the DM2. This subset consists of the concepts described in Table 2. The principal DM2 concepts that will be used in the SIF are Activity, Capability, Information, Performer, and Resource, as described in Table 2.

Concept	Description
Activity	Work, not specific to a single organization, weapon system or individual that transforms inputs (Resources) into outputs (Resources) or changes their state.
Capability	The ability to achieve a Desired Effect under specified [performance] standards and conditions through combinations of ways and means [activities and resources] to perform a set of activities.
Information	Information is the state of a something of interest that is materialized -- in any medium or form -- and communicated or received. Described in Section 2.3.2
Performer	Any entity - human, automated, or any aggregation of human and/or automated - that performs an activity and provides a capability. Described in Section 2.3.3.
Resource	Data, Information, Performers, Materiel, or Personnel Types that are produced or consumed. Described in Section 2.3.1.

Table 2: DoDAF Architecture Concepts

2.3.1 Resources

DoDAF V2.0 defines a Resource as "Data, Information, Performers, Materiel, or Personnel Types that are produced or consumed." The SIF focuses on two categories of Resource: Information and Performers. Information Resources are addressed in the Information Viewpoint. Performers (including systems, services, and users) are addressed in the Computational Viewpoint.

2.3.2 Information Resources

[6] DoD Architecture Framework, http://dodcio.defense.gov/Library/DoD-Architecture-Framework/, Accessed 01 Feb 2017

DoDAF incorporates three levels of information abstraction. These correlate to the three levels commonly used when developing data models for commercial and business applications. These levels are described in Table 3.

Level	Description
Conceptual Data Model	The required high-level data concepts and their relationships.
Logical Data Model	The documentation of the data requirements and structural business process (activity) rules.
Physical Exchange Schema	The physical implementation format of the Logical Data Model entities, e.g., message formats, file structures, physical schema.

Table 3 : Levels of Information Abstraction

Modeling of Information Resources starts with the Conceptual model. This model provides a vocabulary for the concepts required to discuss sensor integration. It describes SIF concepts from the operational or mission perspective, but lacks the specificity needed to guide consistent implementation of those concepts. The SIF-SP allocates the Conceptual Data Model to the Reference View.

The Logical Data Model provides additional information to guide implementation, more complex information than is possible with the concepts alone. It captures the logical associations between the concepts, although these associations are defined only at an abstract level. The Logical model is agnostic to any specific implementing technology. All concepts in the Logical model are traceable to the Conceptual model.

The Logical Data Model is split between the Reference and Technical views of the SIF-SP. In the Reference view, the Logical model must be agnostic to any implementing technology. However, a totally agnostic Logical model is not always sufficient to guarantee coherence with the Physical model. Therefore, if the Reference View Logical Model is insufficient, the Technical View will include an extended Logical model. This extended Logical model provides the additional information necessary for that implementing technology to completely and accurately implement the Reference View. In this way the Logical model serves as an architectural bridge between the Reference and Technical Views

The Physical Exchange Schema provides the requirements to implement the Logical Data Model using a specific technology. It includes schema, lists of valid values, and validation logic. The physical model should have sufficient detail to enable independent and interoperable implementation of the Logical and Conceptual models. The SIF-SP allocates the Physical Exchange Schema to the Technical Views.

2.3.3 Performer Resources

A Performer is a Resource which "performs an activity and provides a capability." They include both Automated and Human Performers. DoDAF Performers are defined in Table 4. The SIF-SP Reference View does not specify performers nor their implementation. Rather, it uses Performers to fill the role of actor for implementing Capabilities and Activities. Technical Views, however, may identify specific performer types and associated standards which are applicable to their implementation environment.

Concept	Definition
Individual Person/Role	Person roles defined by the role or roles they share that are relevant to an architecture. Includes assigned materiel.
Organization	A specific real-world assemblage of people and other resources organized for an on-going purpose.
Organization Type	A type of Organization.
Person/Role Type	A category of person roles defined by the role or roles they share that are relevant to an architecture. Includes assigned materiel.

Concept	Definition
Service	A mechanism to enable access to a set of one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description. The mechanism is a Performer. The "capabilities" accessed are Resources -- Information, Data, Materiel, Performers, and Geopolitical Extents.
System	A functionally, physically, and/or behaviorally related group of regularly interacting or interdependent elements.

Table 4 : Performer Concepts

2.3.4 Capabilities and Activities

In order to achieve a Desired Effect, a Performer must have the Capability to change a Resource to the state required to achieve that Desired Effect. A Capability is composed of Activities, each of which consumes and produces Resources. It is the act of consuming and producing Resources which produces a change to the Resource state.

A Capability is the ability of a Performer to achieve a mission objective. Activities are the steps necessary to execute that Capability. These concepts are defined in Table 5.

Concept	Definition
Activity	Work, not specific to a single organization, weapon system or individual that transforms inputs (Resources) into outputs (Resources) or changes their state.
Capability	The ability to achieve a Desired Effect under specified [performance] standards and conditions through combinations of ways and means [activities and resources] to perform a set of activities.
Desired Effect	A desired state of a Resource
Performer	Any entity - human, automated, or any aggregation of human and/or automated - that performs an activity and provides a capability.
Resource	Data, Information, Performers, Materiel, or Personnel Types that are produced or consumed.
Rule	A principle or condition that governs behavior; a prescribed guide for conduct or action.

Table 5: Capabilities and Activities

2.4 UML Model

The Architecture used in the SIF is developed and validated using a UML model and the Sparx Enterprise Architect modeling tool. While UML models are commonly used by standards developers to model information, the SIF model also models behaviors. The analysis process starts with a set of use cases. Each use case is modeled as a sequence diagram. Each transaction in the sequence diagram must correspond to an operation supported by the target object. This assures that there are no information or control flows which have not been identified and allocated to an entity in the Architecture.

The SIF-SP UML model is a composite of the following packages:

- DoDAF: a subset of the full DoDAF 2.0 UML model
- ISA: a model derived from the ISA 6.0 specifications
- ITU-T: common classes defined by ITU-T standards
- NSG: information and service models based on NSG standards
- ODNI: common classes defined by the ODNI
- OGC: UML models for OGC information and service standards
- W3C: common classes defined by the W3C
- ISO TC211: the harmonized UML model maintained by ISO TC211

- SIF: UML constructs specific to the SIF

Wherever possible, SIF concepts have been imported from their authoritative source. The number of SIF unique concepts have been kept to a minimum.

The engineering rigor that a UML model provides also inhibits the adaptability needed to address changing real-world requirements. This is mitigated by limiting the detail of the information captured in the model. For example, the model defines the Observable class. It does not provide a taxonomy of all possible types of Observable. Such a taxonomy would be out of date before it could be published. Rather, the UML model defines the logical structure and use of an Observable, and leaves the identification of types of Observables to the SIF-SP Ontology.

2.5 OWL Ontology

As an integration framework, the SIF brings together technologies and operational concepts from many communities. It is vital that there be a clear and unambiguous understanding of all SIF concepts. It is also essential that there be a clear mapping between SIF concepts and the concepts of each community. Therefore, the SIF-SP includes the SIF-SP Ontology. This ontology will document the concepts used throughout the SIF as well as the associations between those concepts.

The SIF-SP Ontology is based on four sources: DoDAF, the World Wide Web Consortium (W3C) Semantic Sensor Network (SSN) Ontology, OGC Sensor Web Enablement (SWE), and the Integrated Sensor Architecture (ISA) Data Model.

DoDAF defines a data model for capturing architecture concepts. Since the SIF is primarily an architecture effort, DoDAF concepts form the foundation of the SIF-SP Ontology.

The Semantic Sensor Network has been produced through a collaborative effort between the Open Geospatial Consortium (OGC) and the World Wide Web Consortium (W3C). It serves as an industry standard ontology for the sensor networking domain.

SSN is a fairly small ontology. There are many additional domain concepts required. This gap is addressed by adding concepts from the OGC SWE suite of standards.

Finally, it is necessary to ground the ontology in real systems. As an example, MASBUS and ISA address two very different sensor environments. Concepts from these two environments are used to fill out the SIF-SP Ontology. The result is an ontology which should be recognizable to implementers from both environments.

3 Conformance

The SIF-SP Reference View provides an architecture framework which is agnostic to the implementing technology. As such, conformance with the SIF RV cannot be tested directly. Conformance with this Specification is demonstrated by demonstrating conformance with one of the Technical Views derived from this Specification. Those Technical Views are included under Normative Specifications.

Section 11 discusses interoperability across the defined technical views. The enterprise technical view serves as the common target for data sharing and all other technical views include definitions for bridging information from that view into the enterprise view. Each of the other technical views shall define conformance classes for basic capabilities within the specified communications environment and for bridging to provide interoperability with the enterprise technical view. Compliance with the bridging conformance class is conditionally required when interconnecting beyond the specified communications environment to exchange information in the enterprise communication environment.

The GWS Focus Group (FG) is responsible to confirm and assert that Technical Views derived from this Specification conform to the provisions of the SIF-SP reference view. All Technical Views derived from this Specification shall be established via the governance procedures of the GWS FG and the GWG.

4 Related Specifications

4.1 Normative Specifications

ISO 19107:2003, Geographic information -- Spatial schema, https://www.iso.org/standard/26012.html

ISO 19123:2005, Geographic information -- Schema for coverage geometry and functions, https://www.iso.org/standard/40121.html

ISO 19157:2013, Geographic information -- Data quality, https://www.iso.org/standard/32575.html

DoDI 5105.58 Measurement and Signature Intelligence 22 Apr 2009 -- http://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/510558p.pdf

NGA.SP.0009.02_1.0.1_SIFTENT, *Sensor Integration Framework Standards Profile, Technical View 1 - Enterprise*, 2 August 2019

NGA.SP.0009.04_1.0.1_SIFTDDIL, *Sensor Integration Framework Standards Profile, Technical View 3 - Tactical DDIL IP Environment*, 2 August 2019

SIF-SP Ontology, https://github.com/ngageoint/Sensor_Integration_Framework

SIF-SP UML Model, https://github.com/ngageoint/Sensor_Integration_Framework

4.2 Informative Specifications

DoD Architecture Framework, http://dodcio.defense.gov/Library/DoD-Architecture-Framework/, Accessed 01 Feb, 2017

IETF RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax, August 1998

ISO 19115:2003/Cor.1:2006, Geographic information – Metadata – Corrigendum 1, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=44361ITU-T T.808, JPEG 2000 Interactive Protocol (Part 9 – JPIP), January 2005

ITU-T Rec X.901, Information technology - Open Distributed Processing – Reference Model: Overview, 09 August 1997

ITU-T Rec X.902, Information technology - Open Distributed Processing – Reference Model: Foundations, 29 October 2009

ITU-T Rec X.903, Information technology - Open Distributed Processing – Reference Model: Architecture, 29 October 2009

ITU-T Rec X.904, Information technology - Open Distributed Processing – Reference Model: Architectural Semantics, 12 December 1997

Object Management Group, Unified Modeling Language, version 2.4.1

OGC 06-121, OGC® Web Services Common Implementation Specification v2.0, 7 April 2010
OGC 09-001, OpenGIS® SWE Service Model Implementation Standard v2.0, 21 March 2011

OGC 12-000, OGC® SensorML: Model and XML Encoding Standard v2.0, 4 February 2014
OGC 12-006, OGC® Sensor Observation Service Interface Standard v2.0, 16 April 2012

W3C Recommendation, OWL Web Ontology Language, https://www.w3.org/TR/owl-ref/, Accessed 02 October 2017

5 Terms and Definitions

The SIF-SP Terms and Definitions can be found in 0.

6 Abbreviations

The SIF-SP list of abbreviations can be found in Annex B.

7 Enterprise Viewpoint

The Enterprise Viewpoint is concerned with the purpose, scope and policies governing the activities of the specified system within the organization of which it is a part. In other words, it describes the mission to be accomplished and the operational constraints on that mission. The SIF-SP Enterprise View captures that information through a series of Use Cases (Section 7.1), which are expressed using a set of modeled components (Section 7.2) and data types (Section 7.3) which are realizations of classes from the Computational and Information Viewpoints.

7.1 Use Cases

The Use Cases presented are both generic and sensor agnostic. They serve to illustrate the tie between operational needs and the architecture which addresses those needs. These Use Cases are tightly integrated with the rest of the architecture. This integration provides the ability to trace concepts across Viewpoints and helps assure the consistency and coherence of the SIF model.

7.1.1 Use Case 1: Discover Sensor

Objective: Identify a Sensor which meets the Operator's needs.

1) Operator builds a Query Expression which expresses the criteria for selecting an appropriate Sensor
2) Operator issues a "submitQuery" request to the Service Catalog
3) The Service Catalog accepts the request and selects those Sensors which meet the selection criteria as expressed in the Query Expression
4) The Service Catalog builds a Resource Description Document for each selected Sensor
5) The Service Catalog returns the Resource Description Documents to the Operator
6) Operator receives and evaluates the Resource Description Documents

7.1.2 Use Case 2: Describe Sensor

Objective: Determine the current properties of a sensor.

1) Operator performs Use Case 1, Discover Sensor
2) Operator selects a Sensor and records the unique identifier for that Sensor

3) Operator issues a “describe” request to the Service Catalog
4) The Service Catalog returns a Sensor Description Document to the Operator

7.1.3 Use Case 3: Discover Observations

Objective: Identify the Observations which meets the Operator’s needs.

1) Operator builds a Query Expression which expresses the criteria for selecting appropriate Observations
2) Operator issues a “submitQuery” request to the Observation Catalog
3) The Observation Catalog accepts the request and selects those Observations which meet the selection criteria as expressed in the Query Expression
4) The Observation Catalog builds a Resource Description Document for each selected Observation
5) The Observation Catalog returns the Resource Description Documents to the Operator
6) Operator receives and evaluates the Resource Description Documents

7.1.4 Use Case 4: Describe Observations

Objective: Determine the Sensor which generates desired Observations.

1) Operator performs Use Case 3, Discover Observation
2) Operator selects an Observation and records the unique identifier for that Observation
3) Operator issues a “describe” request to the Observation Catalog
4) The Observation Catalog returns an Observation Description Document to the Operator
5) The Operator evaluates the Observation Description and determines that they want more of these.
6) The Operator extracts the unique identifier for the Sensor which generated that Observation from the Observation Description Document

7.1.5 Use Case 5: Deliver Discrete Measures

Objective: Access Discrete Measures as they are collected by the Sensor.

1) Operator performs Use Case 4, Describe Observation retrieving the unique identifier for a Sensor
2) Operator issues a “describe” request to the Service Catalog
3) The Service Catalog returns a Sensor Description Document to the Operator
4) Operator evaluates the sensor properties and determines a set of better settings
5) Operator issues a “set property” command to the Sensor Manager for each property that is to be modified, passing the new value with each command
6) The Sensor Manager receives each command and:
    a. Validates that the operator is authorized to issue that command
    b. Updates the property of the sensor to the new value.
7) Operator issues a “submitRequest” command to the Observation Server
8) The Observation Server returns the requested Observations to the Operator

7.1.6 Use Case 6: Deliver Streaming Measures

Objective: Stream Full Motion Video (FMV) Measures to the Operator as they are collected by the Sensor.

1) Operator performs Use Case 4, Describe Observation retrieving the unique identifier for a Sensor
2) Operator issues a "describe" request to the Service Catalog
3) The Service Catalog returns a Sensor Description Document to the Operator
4) Operator evaluates the sensor properties and determines a set of better settings
5) Operator issues a "set property" command to the Sensor Manager for each property that is to be modified, passing the new value with each command
6) The Sensor Manager receives each command and:
    a. Validates that the operator is authorized to issue that command
    b. Updates the property of the sensor to the new value.
7) Operator issues a "establishStreamingSession" command to the FMV Server
8) The FMV Server establishes a connection to the Operator's FMV client and begins streaming video.

7.1.7 Use Case 7: Deliver Interactive Streaming Measures

Objective: Stream Measures to the Operator over an interactive interface.

1) Operator performs Use Case 4, Describe Observation retrieving the unique identifier for a Sensor
2) Operator issues a "describe" request to the Service Catalog
3) The Service Catalog returns a Sensor Description Document to the Operator
4) Operator evaluates the sensor properties and determines a set of better settings
5) Operator issues a "set property" command to the Sensor Manager for each property that is to be modified, passing the new value with each command
6) The Sensor Manager receives each command and:
    a. Validates that the operator is authorized to issue that command
    b. Updates the property of the sensor to the new value.
7) Operator issues a "establishInteractiveSession" command to the Imagery Server
8) The Imagery Server establishes a connection to the Operator's Imagery client and begins streaming the image.
9) Operator issues a "pause" command to the Imagery Server over the interactive protocol.
10) The Imagery Server suspends streaming of the image
11) Operator issues a "play" command to the Imagery Server over the interactive protocol
12) The Imagery Server resumes streaming of the image

7.1.8 Use Case 8: Set Sensor Properties

Objective: Determine the current properties of a sensor and adjust them to achieve a better collect.

1) Operator performs Use Case 2, Describe Sensor
2) Operator evaluates the sensor properties and determines a set of better settings
3) Operator issues a "set property" command to the Sensor Manager for each property that is to be modified, passing the new value with each command

4) The Sensor Manager receives each command and:
    a. Validates that the operator is authorized to issue that command
    b. Updates the property of the sensor to the new value.

7.2 Components

Within this Enterprise Viewpoint, components serve the role of Performers which implement Capabilities and execute Activities. They do not define systems that shall or have been built. Nor do they specify the only way the capabilities and activities can be implemented. Those details, where they are appropriate, are provided in the Technical Views. Rather, they serve as a conceptual bridge between the Computational Viewpoint and the Use Cases.

Component	Capability	Activity	Discussion
Service Catalog	Discovery	Browse	
		Describe	
		Register	
		SubmitQuery	
Observation Catalog	Discovery	Browse	
		Describe	
		Register	
		SubmitQuery	
Sensor Manager	Command and Control	Activate	
		Deactivate	
		Get Observation	
		Get Property	
		Get State	
		Process	
		Set Property	
		Task	
		Trigger	
Observation Server	Delivery	SubmitRequest	
FMV Server	Streaming Delivery	EstablishStreamingSession	
		TerminateStreamingSession	
Imagery Server	Interactive Delivery	establishInteractiveSession	
		TerminateInteractiveSession	
Sensor			

Table 6: Use Case Components

7.3 Data Types

Within this Enterprise Viewpoint, data elements provide the additional detail needed to bridge the concepts in the Information Viewpoint with the operational focus of the use cases. Table 7 provides a mapping from Information Viewpoint concepts to the data concepts used in the Use Cases.

Use Case Data Type	Information Viewpoint	Discussion
Resource Description Document	Resource Description	
Observation Description Document	Observation Description	
Sensor Description Document	Performer Description and Activity Description	
Observation	Discrete Measure	
FMV Stream	Measure Stream	
Query Expression	Query Expression	
Image	Interactive Stream	

Table 7: Use Case Data Types

8 Information Viewpoint

The Information Viewpoint provides explanation and relationships between component information entities and concepts. This includes entity or descriptions, observables and observations, properties, spatial-temporal, and messages. The Information Viewpoint describes and characterizes information elements which exist and are exchanged and processed as described in the other Viewpoints.

8.1 Descriptions

Descriptions are information or metadata which describes a resource. This metadata can be defined at two levels of detail. Resource Descriptions provide a general description suitable for any resource. Observable, Observation, Performer, and Activity Descriptions provide descriptions which are specific to those types of Resource. The relations between the Resource Description and other Description types is illustrated in Figure 2.

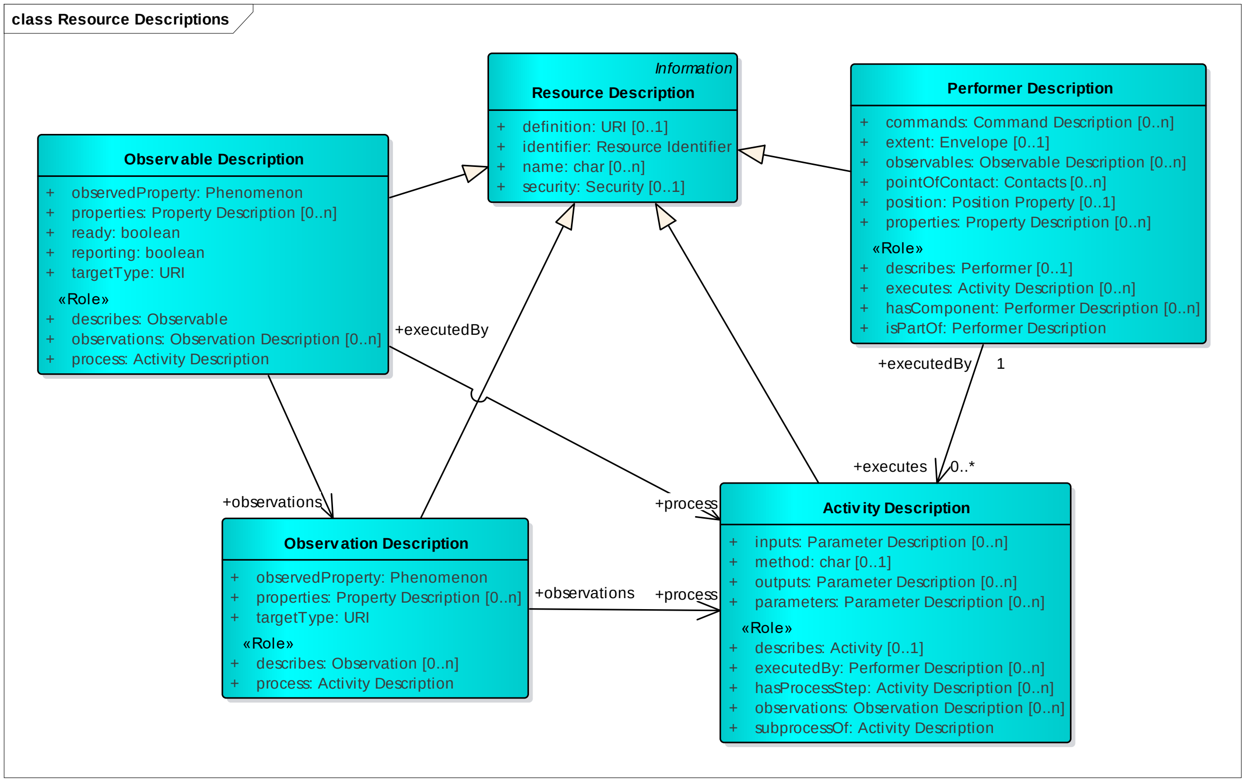


Figure 2 : Descriptions

8.1.1 Resource Descriptions

A Resource Description is a data record which can describe any type of Resource. As such, it is limited to general descriptive concepts such as contact information, resource type, area of coverage, etc.

It's important to remember that there are associations between the different types of Description. For example, an Observable Description will have an association with the Description of the Performer which generates that Observable.

Element	#	Type	Comments
Definition	0..1	URI	A reference to the formal definition of this resource within the SIF-SP Ontology
Identifier	1..1	Resource Identifier	Globally Unique Identifier for this resource
Name	0..n	String	
Security	0..1	Security	May include security markings, Need to Know, digital signatures, etc.

Table 8 : Resource Description

8.1.2 Observable and Observation Descriptions

8.1.2.1 Observable Description

Observables report on the *types* of information that may result from a sensing Activity. In many cases an Observable is simply be an identifier for the type. An Observable Description is metadata which describes a single Observable. Observable Descriptions, Observables, and other related classes are illustrated in Figure 3. Attributes of the Observable Description class are described in Table 9.

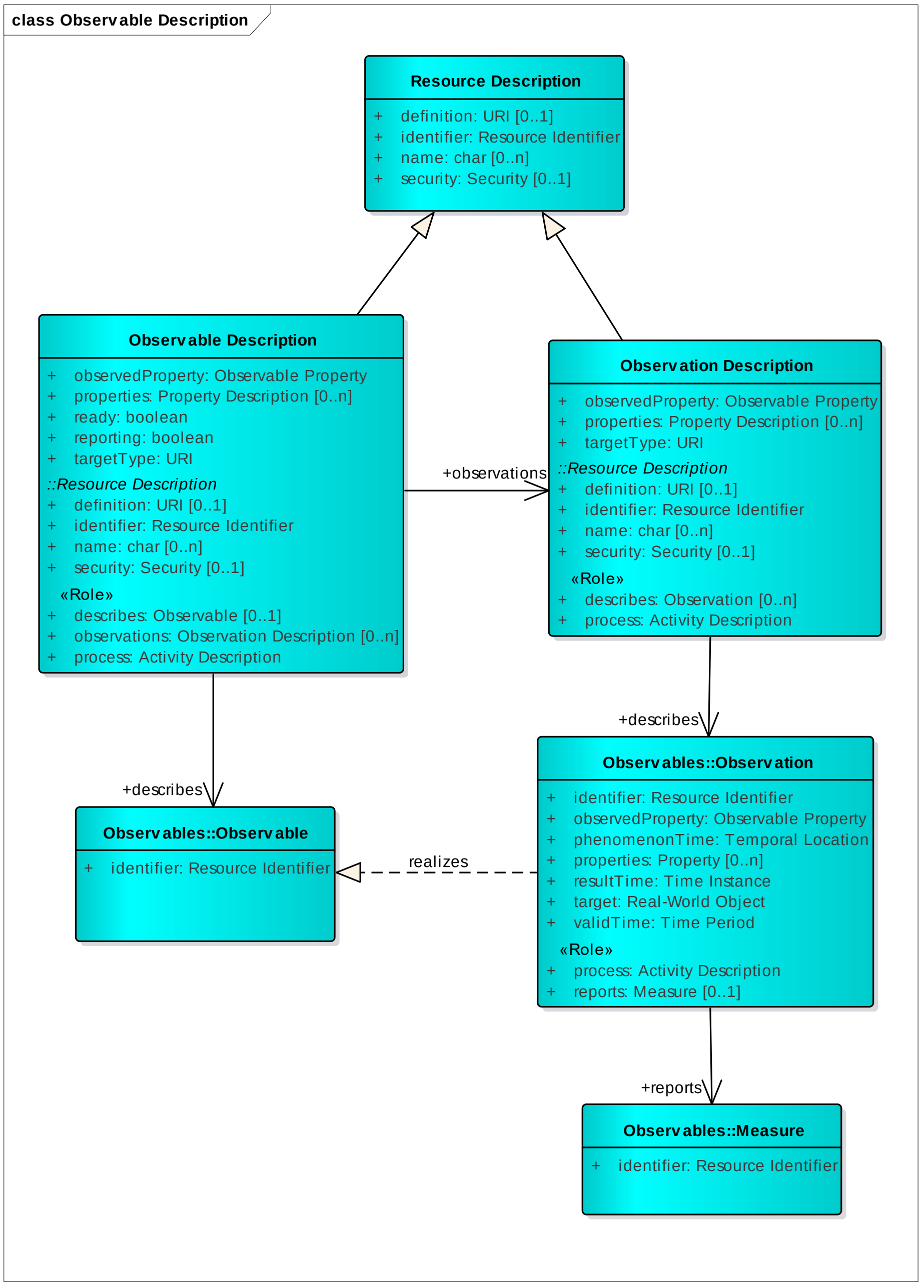


Figure 3 : Observable and Observation Descriptions

Element	#	Type	Comments
Definition	0..1	URI	A reference to the formal definition of this resource within the SIF-SP Ontology
Identifier	1..1	Resource Identifier	Globally Unique Identifier for this resource

Element	#	Type	Comments
Name	0..n	String	
Security	0..1	Security	May include security markings, Need to Know, digital signatures, etc.
observedProperty	1..1	Observable Property	Identifies the property (temperature, color, etc.) that can be measured
Properties	0..n	Property Description	Descriptions of properties which are relevant to understanding or processing these observations.
Ready	1..1	Boolean	Is the process ready to collect observations?
Reporting	1..1	Boolean	Are the observations being reported?
targetType	1..1	URI	Pointer into a registry of real world object types. The combination of target and observed property defines concepts like air (target = atmosphere) temperature (property = temperature)
Describes	1..1	Observable	Pointer to the observable this metadata describes
Observations	0..n	Observation Description	Pointers to metadata for the observations generated for this observable.
Process	1..1	Activity Description	Pointer to metadata describing the Activity which collects the observations.

Table 9 : Observable Description

8.1.2.2 Observation Description

An Observable is a concept representing a type of Observation that may be generated through an Activity. Each Observable Description includes references to Descriptions of each Observation that has been generated. Observation Descriptions are illustrated in Figure 3. The attributes of the Observation Description class are described in Table 10.

Element	#	Type	Comments
Definition	0..1	URI	A reference to the formal definition of this resource within the SIF-SP Ontology
Identifier	1..1	Resource Identifier	Globally Unique Identifier for this resource
Name	0..n	String	
Security	0..1	Security	May include security markings, Need to Know, digital signatures, etc.
observedProperty	1..1	Observable Property	Identifies the property (temperature, color, etc.) that can be measured
Properties	0..n	Property Description	Descriptions of properties which are relevant to understanding or processing these observations.
targetType	1..1	URI	Pointer into a registry of real world object types. The combination of target and observed property defines concepts like air (target = atmosphere) temperature (property = temperature)
Describes	0..1	Observation	Pointer to the observation this metadata describes

Element	#	Type	Comments
Process	1..1	Activity Description	Pointer to metadata describing the Activity which collects the observations.

Table 10 : Observation Description

8.1.3 Performer and Activity Descriptions

At its most fundamental level, a sensor is a performer which executes an activity or process. It receives input, processes that input, and generates output. The processing is defined by an algorithm and controlled through parameters. The processing takes place within the environment defined by the platform upon which the processing is hosted. The platform has properties whose value may influence or contribute to the processing and output formation.

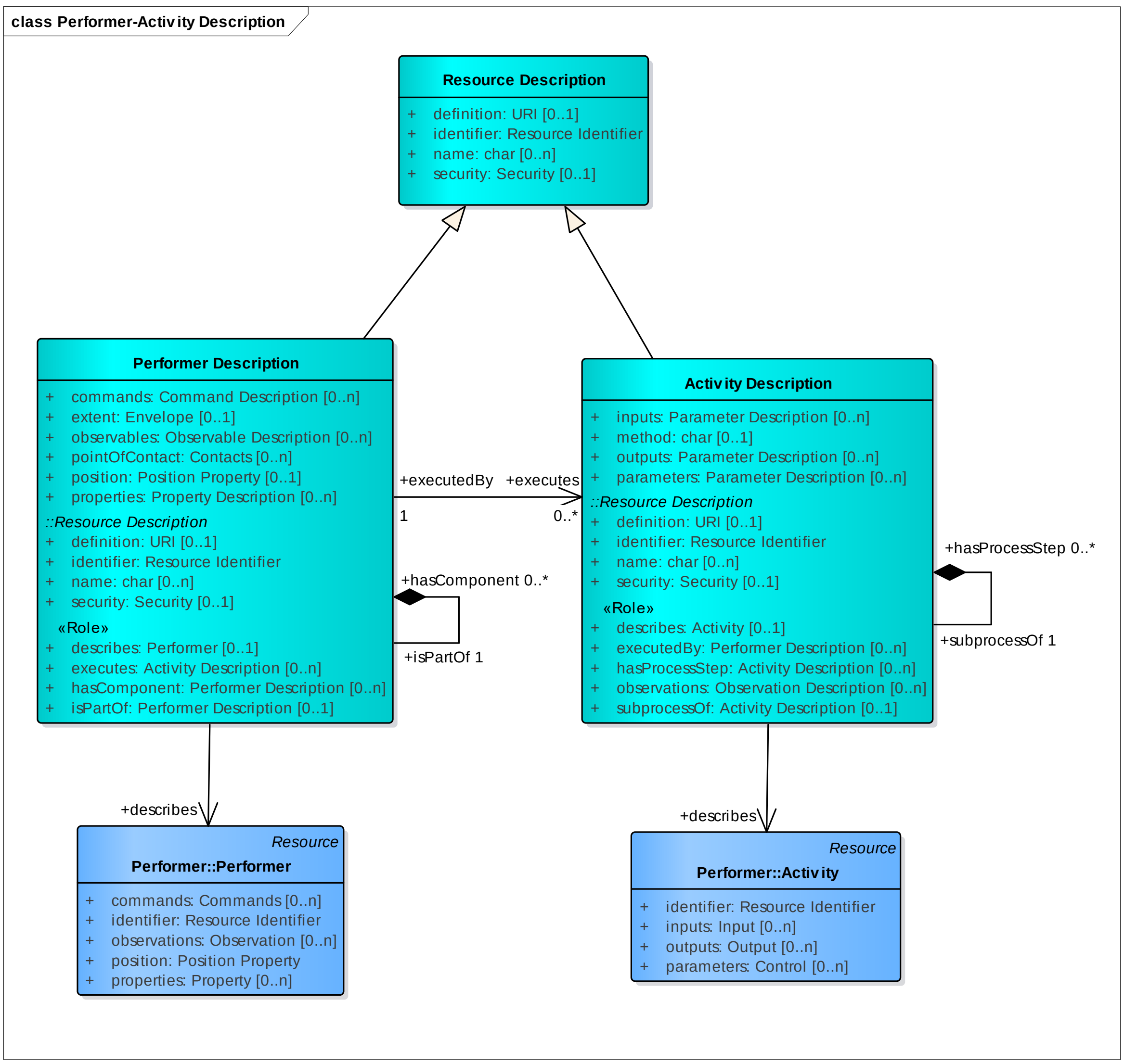


Figure 4 : Performer and Activity Descriptions

8.1.3.1 Performer Description

A Performer is an entity that performs tasks, independent of whether the entity is hardware, software, firmware or wetware. Most SIF Performers will be platforms, sensors, or sensor systems. A Performer Description is metadata which describes a Performer. The Performer Description is illustrated in Figure 4. Attributes of the Performer Description class are described in Table 11.

Element	#	Type	Comments
Definition	0..1	URI	A reference to the formal definition of this resource within the SIF-SP Ontology
Identifier	1..1	Resource Identifier	Globally Unique Identifier for this resource
Name	0..n	String	
Security	0..1	Security	May include security markings, Need to Know, digital signatures, etc.
Commands	0..n	Command Description	Description of the commands supported by this Performer
Extent	0..1	Envelope	The spatial and optionally temporal envelope within which this Performer operates
Observables	0..n	Observable Description	The Observables available from this Performer.
pointOfContact	0..n	Contacts	Contact info
Position	0..1	Position Property	Current location of this Performer in space and time
Properties	0..n	Property Description	Descriptions of the properties of this Performer.
Describes	0..1	Performer	Pointer to the Performer described by this metadata
Executes	0..n	Activity Description	Pointers to the Activities executable on this Performer.
hasComponent	0..n	Performer Description	Pointers to the Performer Descriptions of any sub-components of this Performer
isPartOf	0..1	Performer Description	If this Performer is a sub-component, this is a pointer to the Performer Description for the next higher component.

Table 11 : Performer Description

8.1.3.2 Activity Description

A Performer by itself does not perform sensing. It is the Activities which are executed by the Performer which generates Observations. An Activity be can understood as an Algorithm which executes a standard linear equation of the form y = mx + b where x is the input, m and b are the parameters, and y is the output.

The SIF identifies two forms of Activities. Atomic Activities are Activities which follow a request-response pattern. Processing Activities are Activities which will execute for a period (in some cases unbounded) of time. Metadata describing the Atomic Activities are provided through the Command Descriptions. Command Descriptions are covered in Section 8.1.4. Metadata describing Processing Activities are provided through the Activity Description. Activity Descriptions are illustrated in Figure 4 and described in Table 12.

Element	#	Type	Comments
Definition	0..1	URI	A reference to the formal definition of this resource within the SIF-SP Ontology
Identifier	1..1	Resource Identifier	Globally Unique Identifier for this resource

Element	#	Type	Comments
Name	0..n	String	
Security	0..1	Security	May include security markings, Need to Know, digital signatures, etc.
Inputs	0..n	Parameter Description	Describes the input parameters for the command
Method	0..1	String	Identifies the algorithm used by this activity
Outputs	0..n	Parameter Description	Describes the output values from the command
Parameters	0..n	Parameter Description	Describes parameters which control the execution of the command
Describes	0..1	Activity	Pointer to the Activity this metadata describes
executedBy	0..n	Performer Description	Pointer to metadata describing the Performer who can execute this Activity
HasProcessStep	0..n	Activity Description	Pointers to any sub-processes of this Activity
Observations	0..n	Observation Description	Pointers to metadata describing observations available from this Activity
subprocessOf	0..1	Activity Description	If this is a sub-process, pointer to the parent Activity

Table 12 : Activity Description

8.1.4 Command Description

Performer Descriptions include descriptions of the Commands supported by that Performer. The Command Description class defines the attributes used for that purpose. These attributes are described in Table 13.

Element	#	Type	Comments
Definition	0..1	URI	A reference to the formal definition of this resource within the SIF-SP Ontology
Inputs	0..n	Parameter Description	Describes the input parameters for the command
Name	1..1	String	Name of the command
Outputs	0..n	Parameter Description	Describes the output values from the command
Parameters	0..n	Parameter Description	Describes parameters which control the execution of the command
Ready	1..1	Boolean	Is the command ready to execute?

Table 13 : Command Description

8.1.5 Parameter Description

Activity Descriptions and Command Descriptions include descriptions of the Inputs, Outputs, and control Parameters accepted by that Activity or Command. The Parameter Description class defines the attributes used for that purpose. These attributes are described in Table 14.

Element	#	Type	Comments
Constraints	0..1	String	Specifies the valid values for this parameter
Control	1..1	Boolean	Is this parameter a control parameter?

Element	#	Type	Comments
Definition	0..1	URI	A reference to the formal definition of this parameter within the SIF-SP Ontology
Input	1..1	Boolean	Is this parameter an input parameter?
Name	1..1	String	Name of the parameter
Output	1..1	Boolean	Is this parameter an output parameter?
Type	1..1	String	Describes the structure or data type of the parameter.

Table 14 : Parameter Description

8.1.6 Property Description

Performer Descriptions include descriptions of the Properties possessed and reportable by that Performer. These Properties may also be described through the Observable and Observation Descriptions. The Property Description class defines the attributes used for that purpose. These attributes are described in Table 15.

Element	#	Type	Comments
Definition	0..1	URI	A reference to the formal definition of this property within the SIF-SP Ontology
Name	1..1	String	Name of the property
readOnly	1..1	Boolean	Indicates that this property cannot be set by the user.
ready	1..1	Boolean	Indicates that this property is populated
reporting	1..1	Boolean	Indicates that this property value is being reported
type	1..1	String	Describes the structure or data type of the property.

Table 15 : Property Description

8.2 Component Properties

Component Properties represent dynamic and static properties of a Performer or Activity. The SIF organizes these Properties into a taxonomy to better convey their meaning and purpose. This taxonomy identifies nine (9) categories of properties. Each is intended to provide scope as to the purpose and potential use of each Properties in each category. The taxonomy of Properties is illustrated in Figure 5.

This taxonomy is provided for sole purpose of clarifying the meaning and use of individual Properties. There is no SIF requirement that the taxonomy must be reflected in SIF-SP implementations.

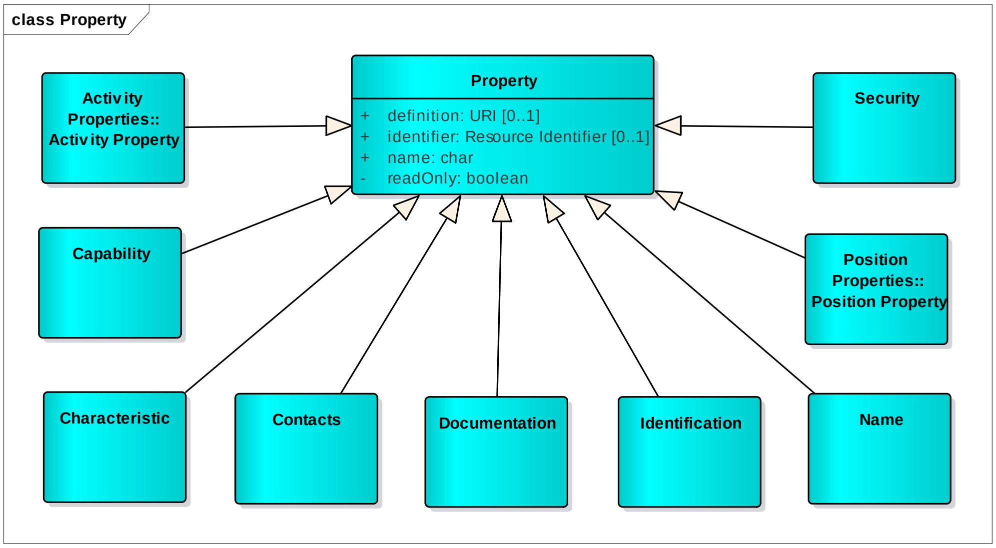


Figure 5 : SIF Component Properties

8.2.1 Properties

The Properties class is the root class of the Component Properties. All Properties implement the attributes defined for this class. In addition, any Property which does not fit in one of the categories of the taxonomy should be implementations of this class. The attributes of the Properties class are described in Table 16.

Element	#	Type	Comments
Identifier	1..1	Resource Identifier	Unique identifier for this Property
Definition	0..1	URI	A reference to the formal definition of this property within the SIF-SP Ontology
Name	1..1	String	Name of the Property
readOnly	1..1	Boolean	Indicates that this property cannot be set by the user

Table 16 : Properties Class Attributes

8.2.2 Activity Properties

Activity Properties are those properties which are associated almost exclusively with Activities. Their primary purpose is to provision and control the execution of the activity. The Activity Properties are illustrated in Figure 6.

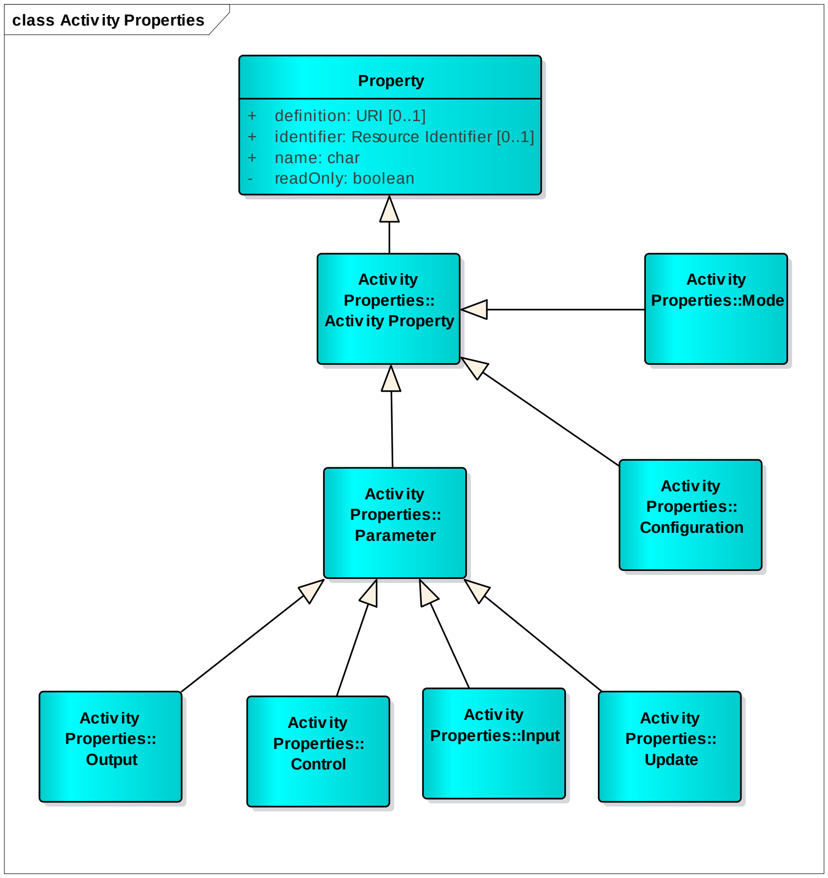


Figure 6 : Activity Properties

8.2.2.1 Parameter

Parameters are the list of data components (and their properties and semantics) that the Activity will accept as parameters. They should be understood in the context of a standard linear equation of the form y = mx + b where:

x = the input,

m = a control,

b = a control,

y = the output.

These Parameters are described in Table 17.

Parameter	Description
Control	A data item or reference to a data item which is neither input nor output but will be used to direct the processing of a command or activity.
Input	A data item or reference to a data item to be used as input to a command or activity.
Output	Identifier for or reference to a location where the output of a command or activity is to be stored.

Parameter	Description
Update	A reference to a data item which will serve as both input and output to a command or activity. The data item will be changed (updated) by the command or activity.

Table 17 : Activity Parameter Properties

8.2.2.2 Configuration

Configuration Properties are value settings which are not specific to any execution of the Activity but do server to further constrain the execution of that Activity.

8.2.2.3 Mode

Modes are Properties which represents a collection of parameters that can be set at once through the selection of a particular predefined mode.

8.2.3 Capability

Capabilities are Properties that further clarify or quantify the output (Observations) of an Activity. Examples of Capabilities include dynamic range, sensitivity, threshold, etc. These properties assist in the discovery of processes that meet particular requirements.

8.2.4 Characteristic

Characteristics are Properties which, while useful, do not further qualify the output (Observations) of an Activity. Examples of Characteristics include component dimensions, battery life, operational limits, etc.

8.2.5 Contacts

Contacts Properties identify individuals and organizations responsible for a resource. They also provide information on how these responsible parties can be contacted. Contacts Properties are based on CI_ResponsibleParty class from ISO 19115.

8.2.6 Documentation

Documentation Properties identify additional external online documentation of relevance to a component. Examples include user's guides, product manuals, specification sheets, images, technical papers, etc.

8.2.7 Identification

Identification Properties are identifiers which are useful for the discovery of a Resource. Examples include short name, mission id, wing id, serial number, etc.

8.2.7.1 Classifier

Classifier Properties are a special type of Identification Property. These Properties are populated with an identifier from a managed taxonomy of Resource types. This aids in the rapid discovery or organization of processes, sensors, or sensor systems. For example, a taxonomy of CBRN sensor types would greatly speed the discovery of available CBRN sensors.

8.2.8 Name

The Name properties provide a label or identifier for the Resource, commonly a descriptive name. A Resource may have several names, typically assigned by different authorities. In common usage there will be one name per authority, so a processing application may select the name from its preferred codeSpace.

8.2.9 Position Properties

Position Properties provide information about the component's spatial-temporal properties in respect to an external reference frame. Positional information can be given by location, by full body state, by a time-tagged trajectory, or by a measuring or computational process. The SIF Spatial-Temporal Properties are illustrated in Figure 7.

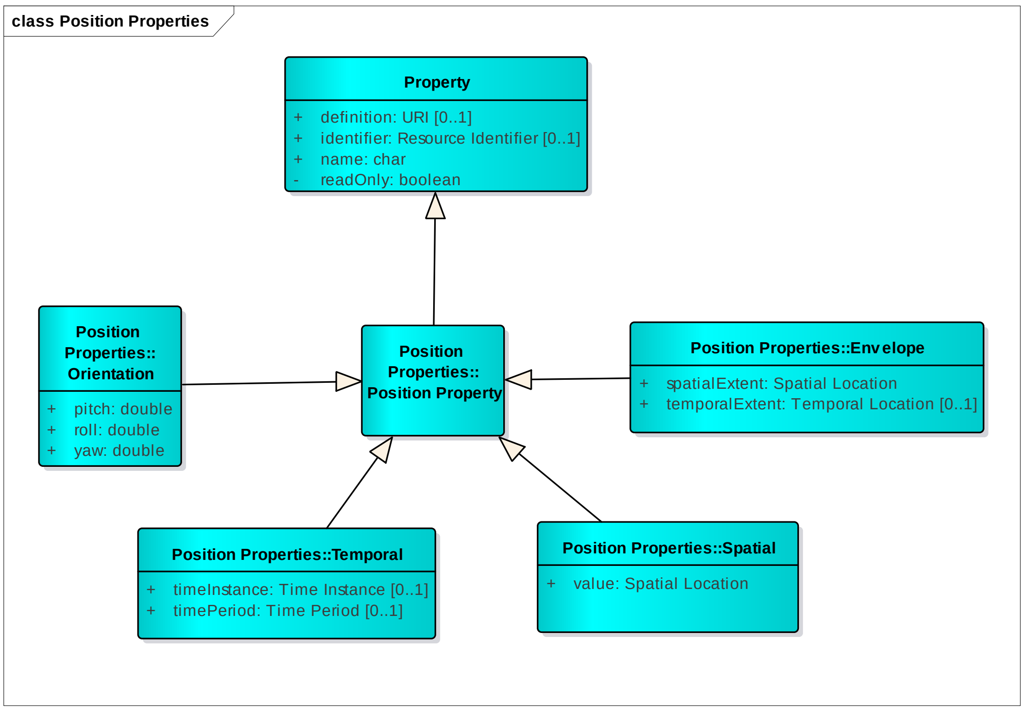


Figure 7 : Spatial-Temporal Properties

8.2.9.1 Envelope

The Envelope Property supports discovery of a resource using spatial and temporal bounds. It describes the minimum spatial box or polygon that encloses the entire resource. Optionally, it can specify the temporal period within which the resource is valid. The attributes of the Envelope Property Class are described in Table 18.

Element	#	Type	Comments
spatialExtent	1..1	Spatial Location	The area within which a Resource is bounded
temporalExtent	0..1	Temporal Location	The time period within which a Resource is bounded

Table 18 : Envelope Properties

8.2.9.2 Orientation

The Orientation Property describes the rotational relationship of an object relative to an external coordinate system. It is typically expressed by relating the rotation of an object's local coordinate axes relative to those axes of an external reference coordinate system. The attributes of the Orientation Property Class are described in Table 19.

Element	#	Type	Comments
pitch	1..1	Double	Rotation around an axis perpendicular to the surface of the earth and perpendicular to the direction of movement.
yaw	1..1	Double	Rotation around the vertical axis.
roll	1..1	Double	Rotation around the axis in the direction of movement.

Table 19 : Orientation Properties

8.2.9.3 Spatial

Spatial Properties define the geometry and location of a Resource. Spatial Properties have a single attribute as described in Table 20.

Element	#	Type	Comments
value	1..1	Spatial Location	A location and associated geometry for a Resource.

Table 20 : Spatial Properties

8.2.9.4 Temporal

Temporal Properties specify the location of a Resource in time. The attributes of the Temporal Property Class are described in Table 21.

Element	#	Type	Comments
timeInstance	0..1	Time Instance	Temporal location as a point in time
timePeriod	0..1	Time Period	Temporal location as a period

Table 21 : Temporal Properties

8.2.10 Security

Security Properties provide for the tagging of a Resource with security related metadata. This property includes security markings, access control lists, digital signatures, and any other markings appropriate for protecting the confidentiality and integrity of a resource.

8.3 Observables and Observations

Observables, Observations, and Measures represent the results of a sensing activity in three levels of abstraction. Observables describe what types of sensor data can be reported. Measures are instances of reported data. An Observation associates an Observable with a Measure. An Observation also contains additional information about when and how the Measures were acquired. The associations between Observables, Observations, Measures, and their descriptions are illustrated in Figure 8.

An understanding of Observables, Observations, and Measures requires an understanding of Phenomenon, Real-World Objects, and Observable Properties.

The OGC SensorML standard defines a phenomenon as “A physical state that can be observed and its properties measured.” As an abstract concept this definition is fine. As the basis for a software design it leaves a little to be desired. More specific concepts are required.

An obvious concept to consider are the “properties measured”. Examples include temperature, speed, color, or just any other property which can be observed and measured. The Observable Property concepts represents the class of properties that can be observed and measured.

But to measure something, it must exist in the real world. Autonomous properties do not exist in the real world. Nor, for that matter, do autonomous states. To fully define a phenomenon, it is necessary to tie it to something that does exist in the real world. A Real-World Object.

Therefore, a phenomenon is described in terms of the Observable Property (or Properties) that can be measured and the Real-World Object whose properties are the subject of that measurement.

Table 22 illustrates how Observables, Observations, and Measures report on Real-World Objects, Observable Properties, and the quantitative measures of those properties.

	Observables	**Observations**	**Measures**
Real-World Object Type	X		
Real-World Object		X	
Observable Property	X	X	
Quantitative Measures		Optional	X

Table 22 : Observables, Observations and Measures

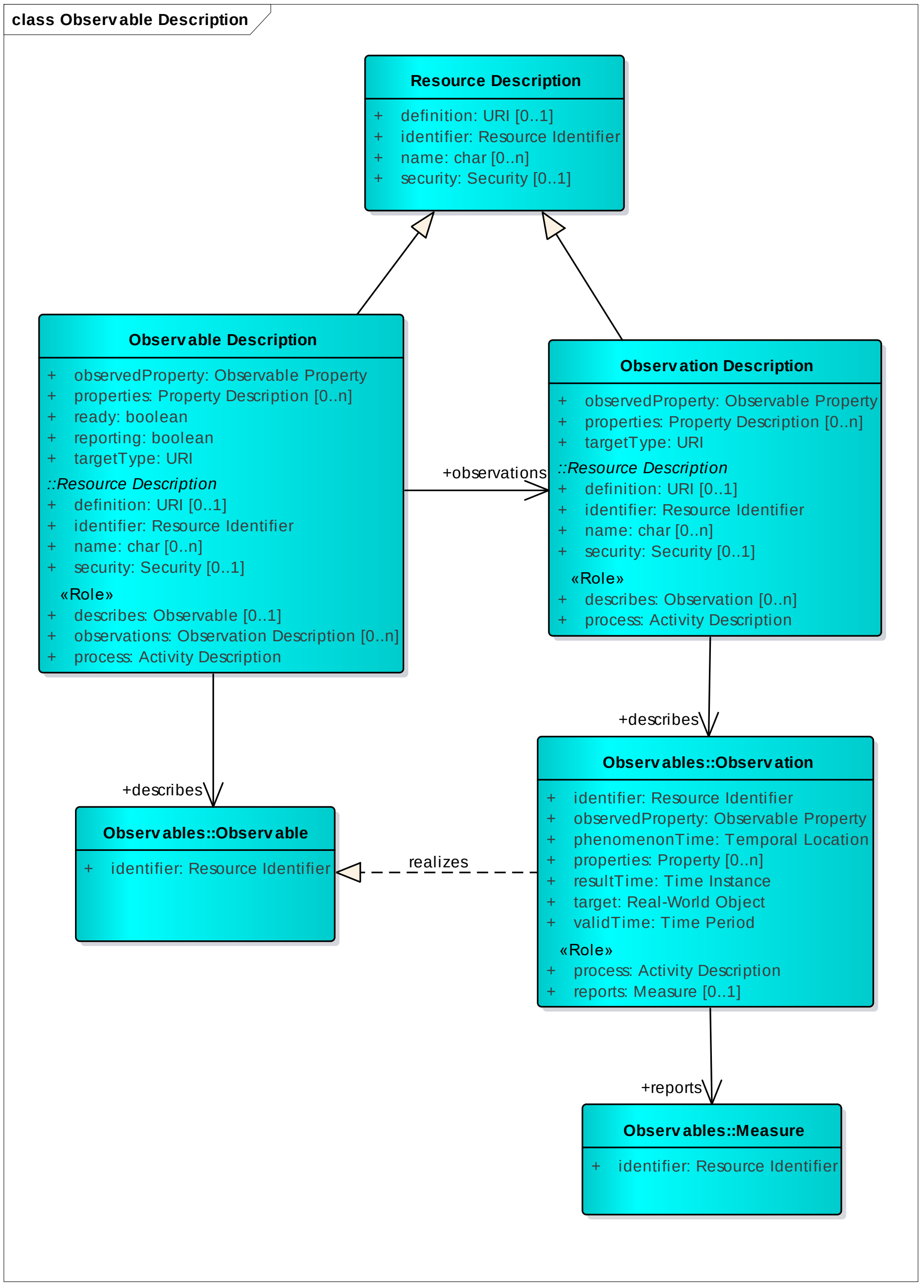


Figure 8 : Observables, Observations, and Measures

8.3.1 Observables

Observables are Information types which represent the types of detections that a sensor or sensor system are capable of collecting. An Observable should convey both the Observable Properties and the types of Real-World Objects that can be reported on. Depending on the implementing environment, this information may be included in the Observable itself or maintained separately and referenced by the Observable identifier. The Observables are described in Table 23.

Element	#	Type	Comments
Identifier	1..1	Resource Identifier	Globally Unique Identifier for this observable

Table 23 : Observables

8.3.2 Observations

Observables are realized through Observations. An Observation includes the descriptive information from the Observation as well as information specific to a sampling event. That information includes the time the measurement took place, the time the Observation was generated, and a time period during which this Observation can be considered valid. In addition, the Observation may also include a set of parameter values which capture the state of the collecting system at the time the Observation was made. Observations are described in Table 24.

Element	#	Type	Comments
Identifier	1..1	Resource Identifier	Globally Unique Identifier for this observation
observedProperty	1..1	Observable Property	See Section 8.3
phenomenonTime	1..1	Temporal Location	When the measured phenomenon took place. May be a time instance or a time period.
properties	0..n	Property	Properties important for the correct interpretation and processing of this Observation
resultTime	1..1	Time Instance	When this observation was generated
target	1..1	Real-World Object	See Section 8.3**Error! Reference source not found.**
validTime	1..1	Time Period	Time period during which this observation will be valid. Default start time is the result time.
Process	1..1	Activity Description	Pointer to the metadata record for the Activity which generated this Observation
reports	0..n	Measure	Pointer to the measures resulting from this measurement event.

Table 24 : Observations

8.3.3 Measure

The primary purpose of a sensor system is to estimate the value of a natural phenomenon. The digital representation of that estimate is a Measure. Just as there are many different types of phenomenon, there are many ways of representing phenomenon digitally. The SIF taxonomy of Measure types is illustrated in Figure 9. The Measure Types are:

- **Coverage:** A Coverage is a homogeneous collection of Discrete Measures which are contained within a spatial boundary. A coverage can be either a regular grid (such as an image) or an irregular grid (such as a point cloud).
- **Discrete Compound:** Discrete Measures are single measurements taken at a specific date and time. Compound Measures are Discrete Measures composed of multiple Scalar and Compound values.
- **Discrete Scalar:** Discrete Measures are single measurements taken at a specific date and time. Scalar Measures are Discrete Measures represented using a standard data type such as Integer, Double, Text, etc.

- **Interactive Stream:** An Interactive Stream provides an incremental delivery of the collected data. The specific content and resolution of that content is specified by the recipient. This allows a user to browse and explore a large collection without having to receive the full Observation. JPIP is an example of Interactive Streaming.
- **Measure Stream:** Stream Measures are a continuous sequence of time-synchronized data capturing the state of the Observed Property over time. Common examples are video and audio streaming.

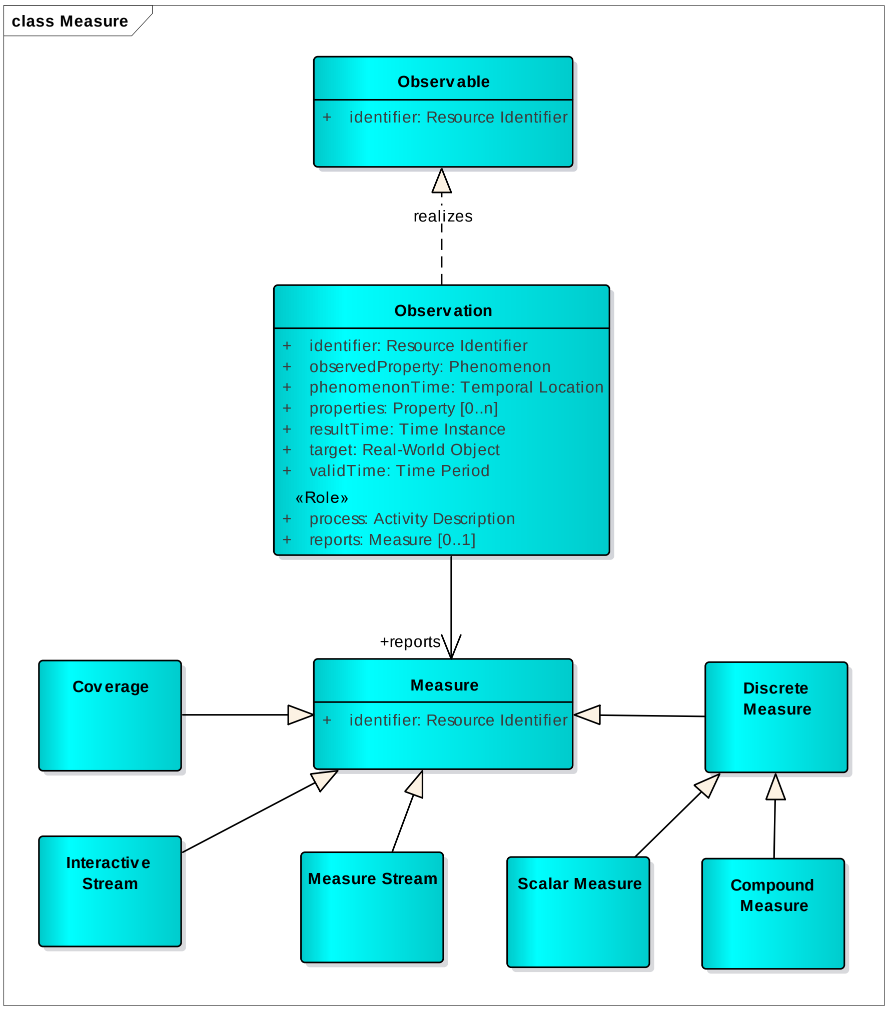


Figure 9 : Measures

8.4 Spatial-Temporal

Almost every discipline uses the concepts of location in space and time. It is only logical that in the interest of interoperability, these concepts should have a consistent meaning and representation. What is not so obvious is that these concepts can be incredibly complex. For example, measuring the distance between two points would seem to be a simple process, but there are a few variables which can cause problems:

1) One location is in decimal degrees and the other in degrees, minutes, and seconds.
2) One location is on the ellipsoid and one on the geoid.

3) The locations are on a projection which does not preserve distance.

And there are many more. Therefore, the spatial-temporal concepts used in the SIF are based on the standards developed by the Open Geospatial Consortium (OGC) and ISO Technical Committee 211. In recognition of the constraints imposed by Size Weight and Power (SWAP) these concepts have been simplified for use in small tactical systems. The simplifying assumptions are included in the descriptions below.

The spatial and temporal concepts supported by the SIF are illustrated in Figure 10 and described in the following sections.

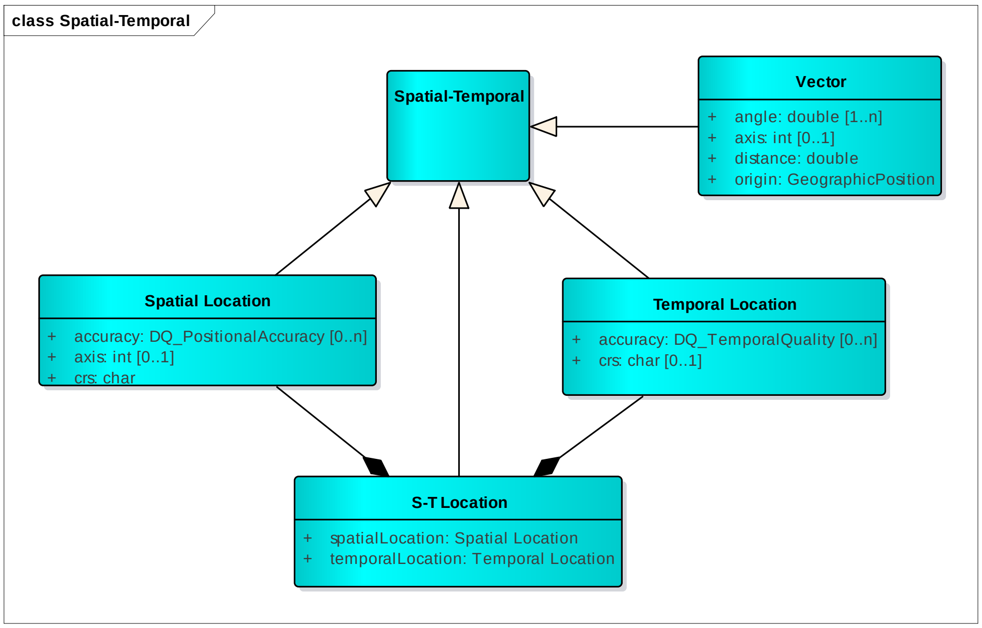


Figure 10 : Spatial-Temporal

8.4.1 Spatial Location

Spatial Locations are shapes and locations in an n-dimensional space. The spatial concepts used in the SIF have an Earth-centric (Geographic) flavor. However, a careful reading of the definitions will show that they can be applied to any spatial reference system of any number of dimensions. A Geographic Position may be a location on the surface of the Earth. Or just as readily a mount point on a satellite chassis. The Spatial Location classes are illustrated in Figure 11. The attributes of the Spatial Location class (which are inherited by all of its subclasses) are described in Table 25.

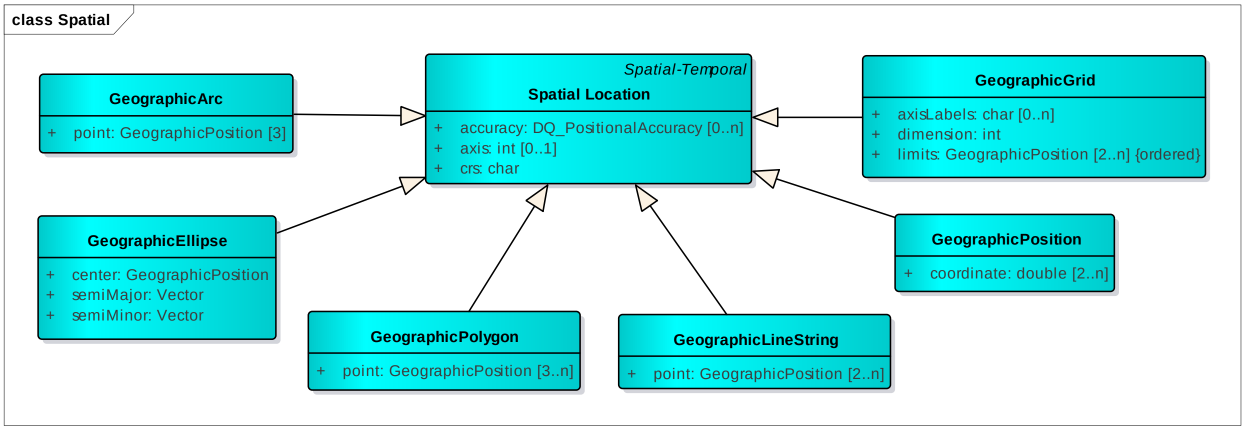


Figure 11 : Spatial Locations

Element	#	Type	Comments
Accuracy	0..n	DQ_PositionalAccuracy	Positional accuracy as defined by ISO 19157. Also available from the NSG registry at https://nsgreg.nga.mil/ir/registers.jsp?register=QM
Axis	0..1	Integer	The number of dimensions. This value can be calculated from the CRS if needed.
CRS	1..1	Char	Identifier for the coordinate reference system used by this element. The value should be an EPSG identifier.

Table 25 : Spatial Location Class Attributes

8.4.1.1 GeographicArc

Implements ISO 19107 GM_Arc

A GeographicArc is a line drawn between three points with a smooth interpolation between the points. The result is a curved line with no visible nodes except for the start and end node.

Element	#	Type	Comments
point	3..3	GeographicPosition	

Table 26 : Geographic Arc

8.4.1.2 GeographicEllipse

A GeographicEllipse is a closed shape with the following properties:

1) The shape is constructed around a center point
2) The boundary of the shape uses smooth interpolation like an Arc.
3) The shape is defined by a major (longest) and minor (shortest) axis which passes through the center point
4) The major and minor axis are perpendicular to each other

The attributes of the GeographicEllipse are described in Table 27.

Element	#	Type	Comments
center	1..1	GeographicPosition	
semiMajor	1..1	Vector	
semiMinor	1..1	Vector	

Table 27 : Geographic Ellipse

8.4.1.3 GeographicGrid

GeographicGrid is based on ISO 19123 CV_Grid

The ISO 19123 concept of a Grid is complex. Much more complex than is needed for the SIF. Therefore, the SIF GeographicGrid class is a simplified, but compatible form of the CV_Grid.

In its simplest form, a grid is a boundary whose interior is populated by a homogeneous collection of objects. An image, for example, is a collection of pixels contained within the image boundaries. However, a grid is not limited to two dimensions. A hypercube (an "image" generated by a hyperspectral sensor") is a grid with three dimensions. In meteorology there are grids which go to five, even seven dimensions. Which leads us to the two mandatory attributes of the GeographicGrid class:

- Dimensions: the number of dimensions in the grid boundary
- Limits: the GeographicPositions which define the grid boundary

It's important to note that only the geometry of the grid boundary is described. The type and ordering of the objects contained within the grid are not defined. That is left to the higher-level object which incorporates the GeographicGrid geometry.

The final attribute is the optional axisLabels attribute. In many grids there is an explicit meaning associated with each axis. For example, in a 3-D grid is would be useful to know which axis is latitude, which is longitude, and which is elevation. The axisLabels attribute provides that information by aligning a text descriptor with each axis.

Element	#	Type	Comments
axisLabels	0..n	String	Provides a list of labels for the axes of the grid The number of labels, if provided, must match the number of dimensions
Dimension	1..1	Integer	Specifies the number of dimensions in the grid. Note that this is not the same as axis. For example, a 2D grid may have 3D elements. Dimension = 2, axis = 3.
Limits	2..n	GeographicPosition	Specifies the boundaries of the grid. The minimal case is a bounding box defined by two coordinates indicating a diagonal across the grid.

Table 28 : Geographic Grid

8.4.1.4 GeographicLineString

A LineString is defined by two or more coordinates with linear (straight line) interpolation between them.

Element	#	Type	Comments
point	2..n	GeographicPosition	The points which make up the line string listed in sequence as you progress down the string.

Table 29 : Line String

8.4.1.5 GeographicPolygon

A Polygon defines an area which is a single surface. The boundary of the surface is defined by a sequence of coplanar points. Planar interpolation is used to define points within the interior of the Polygon.

Element	#	Type	Comments
point	3..n	GeographicPosition	One point for each vertex of the polygon. The points must be ordered so that they progress in a counter clockwise rotation. Failure to do so may result is a polygon which encompasses everything outside of the boundary

Table 30 : Geographic Polygon

The points in a polygon can be listed in either a clockwise or counter clockwise rotation. For the SIF we specify counter clockwise. This ordering determines which side of the boundary is the interior of the polygon and which is the exterior. In most cases this makes little difference. However, with dealing with holes, or polygons which cross a hemisphere boundary, the results of an incorrect ordering can be surprising.

8.4.1.6 GeographicPosition

GeographicPosition implements ISO 19107 GM_Point. It is defined by a single coordinate tuple.

Element	#	Type	Comments
coordinate	2..n	Double	A single location in an arbitrary number of dimensions. The number of coordinates = the number of dimensions.

Table 31 : Geographic Position

8.4.2 Temporal Location

The Temporal Location class is the root class for classes which specify a location (time) or distance (time period) in a temporal reference system. The Temporal Location class is illustrated in Figure 12 and described in Table 32.

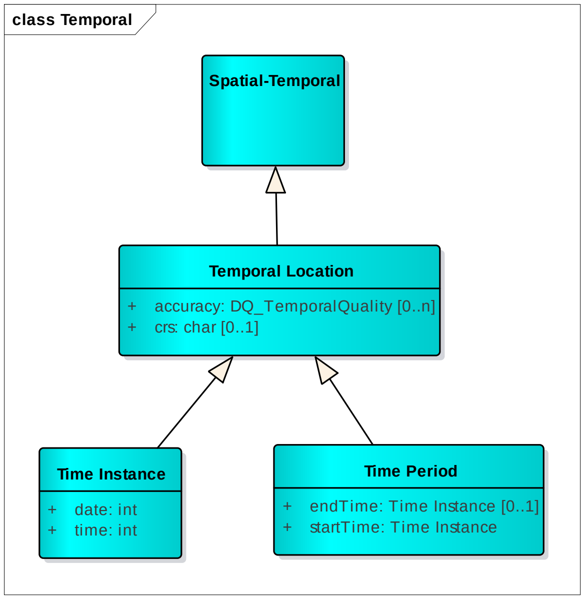


Figure 12 : Temporal Location

A location in space or time is specified in respect to a datum and units of measure. For example, POSIX time is a measure of the seconds elapsed since 0:00:00 January 1, 2017. In this case the temporal datum is January 1, 1970 and the units of measure is seconds. The datum and units of measure are not specified in the Temporal Location class but can be derived from the temporal Coordinate Reference System.

Element	#	Type	Comments
Accuracy	0..n	DQ_TemporalQuality	Temporal quality as defined by ISO 19157. Also available from the NSG registry at https://nsgreg.nga.mil/ir/registers.jsp?register=QM
CRS	0..1	Char	Identifier for the coordinate reference system used by this element. Authoritative temporal CRS definitions are still to be identified.

Table 32 : Temporal Location Class Attributes

8.4.2.1 Time Instance

The Time Instance class identifies a specific point in time. This model is conceptual. Alternate forms are allowed as long as they are mathematically convertible to this form without any loss of precision or accuracy. For example, POSIX time is an integer count of the seconds since January 1, 1970. Conversion of POSIX time into data and time values is a well understood algorithm.

The Time Instance class is described in Table 33.

Element	#	Type	Comments
Date	1..n	Integer	
Time	1..1	Integer	

Table 33 : Time Instance Class Attributes

8.4.2.2 Time Period

The Time Period class indicates a time interval defined by a start time and an end time. The Time Period class does not indicate if the interval is inclusive or exclusive. At this point no requirement has been identified for this degree of specificity. The Time Period class is described in Table 34.

Element	#	Type	Comments
StartTime	1..1	Time Instance	Earliest boundary for the time period
EndTime	0..1	Time Instance	Latest boundary for the time period. If not supplied, then the EndTime is the same as the StartTime.

Table 34 : Time Period Class Attributes

8.4.3 Vector

The Vector class conveys the direction and distance from an origin location to a remote location. This particular class uses rotation angles to define the direction to the remote location. Other techniques are allowed as long as they are mathematically convertible to this method without any loss of precision or accuracy. The Vector class is described in Table 35.

Element	#	Type	Comments
Angle	0..n	Double	Angle provides the angle turns from the neutral axis required to align with the desired direction. Rotations are in-order in a non-rotating frame of reference. The number of angles is the number of axis -1.
Axis	0..1	Integer	Number of dimensions traversed by this vector. Default value = 2. May be derived from context in which it is used.
Distance	1..1	Double	Length of the vector. Note this this does not change with rotation.
Origin	1..1	GeographicPosition	The origin of the vector.

Table 35 : Vector Class Attributes

8.4.4 Space-Time Location

The Space-Time Location class combines a spatial location and a temporal location in a single class. This allows an Activity to identify a unique location in time and space. As a combined element, there is no ambiguity about the scope of the temporal location vs. the scope of the spatial location. The attributes of the Space-Time Class are provided in Table 36.

Element	#	Type	Comments
spatialLocation	1..1	Spatial Location	Spatial Location at the specified Temporal Location
temporalLocation	1..1	Temporal Location	Temporal Location at the specified Spatial Location

Table 36 : Space-Time Location Class Attributes

9 Computational Viewpoint

The Computational Viewpoint attempts to answer the question “what can it do?” There is no easy way to answer this question since it depends on the level of detail desired. DoDAF provides us with three tools to answer this question:

1) Capabilities: What does it do?
2) Activities: What specific actions does it take?

3) Performers: Who or what does the work?

In this Computational Viewpoint of the SIF-SP we will explore how each of these tools are employed to define the Sensor Integration Framework.

9.1 Capabilities

DoDAF defines Capabilities as "The ability to achieve a Desired Effect under specified [performance] standards and conditions through combinations of ways and means [activities and resources] to perform a set of activities."[7] A suitably obscure definition. The U.K Ministry of Defense (MOD) definition is a bit more useful "Capabilities in the MODAF sense are specifically not about equipment but are a high-level specification of the enterprise's ability. A capability is a classification of some ability – and can be specified regardless of whether the enterprise is currently able to achieve it. For example, one could define a capability "Manned Interplanetary Travel" which no-one can currently achieve, but which may be planned or aspired to. Capabilities in MODAF are not time-dependent – once defined they are persistent. It is only the Capability Requirement that changes."[8] For purposes of the SIF, it is sufficient to understand that a Capability is the <u>ability</u> to do something. It does not address <u>how</u> it is to be done.

As a matter of convenience, the SIF organizes Capabilities into six Capability Groups. Each group is described in the sections below.

9.1.1 Messaging

Messaging is the category of Capabilities which allows packets of information (messages) to be exchanged between components. Messaging is part of the infrastructure and arguably should not be in the Reference View. However, the message passing techniques implemented in a deployment environment have a big impact on the behavior of that environment. A common understanding of these techniques is an important part of mapping behaviors between the Technical Views.

The Messaging category contains three Capabilities; Direct Messaging, Message Oriented Middleware, and Publish-Subscribe (see Figure 13).

---

[7] DoD Deputy Chief Information Officer, *DoDAF DM2 Data Dictionary and Mapping*, http://dodcio.defense.gov/Portals/0/Documents/DODAF/DM2_Data_Dictionary_and_Mappings_v202.xls; accessed August 10, 2017,

[8] ibid

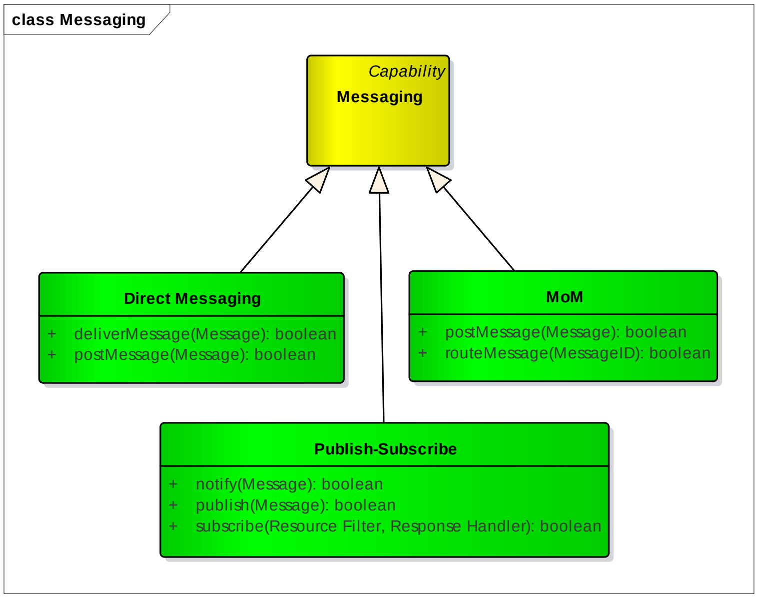


Figure 13 : Messaging Capabilities

9.1.1.1 Direct Messaging

Direct messaging is the Capability for a user to send a message to a specified recipient or group of recipients. Common examples are e-mail and Instant Messaging. Direct messaging is also found in the Request-Response message pair used in the Remote Procedure Call (RPC) design pattern. Web Services are a common example of the RPC pattern.

9.1.1.2 Publish-Subscribe

Publish-Subscribe Messaging is the Capability for a user to broadcast a message which can be received by anyone who may have an interest. Key to this Capability is the Publish-Subscribe service which matches published messages with interested subscribers. This matching is achieved through the use of a filtering algorithm that evaluates the selection criterial provided by the subscriber against the content and metadata of the message. Critical to the effective use of Publish-Subscribe is a well-defined standard for Messages, Metadata, and Filter Language.

9.1.1.3 Message Oriented Middleware

Message Oriented Middleware is a Messaging Capability where the routing decisions are made by the messaging infrastructure, not the sender or recipient. One example of this is an Orchestration Service. In an orchestrated workflow, all information routing is performed by the orchestration engine based on the information received and the business logic encoded in a workflow script.

9.1.2 Discovery

Discovery is the first, and arguably the most important, group of capabilities. It includes the capabilities for users to initially discover, and then evaluate the resources which are available. Discovery is often an iterative process. It begins with a simple query against a set of metadata cards. Similar to the card catalog at a library. The Discovery capability evaluates the query expression against the metadata cards and selects those cards which meet the query criteria. The returned metadata allows the user to build more refined queries, yielding richer metadata. At some point in the process, the user begins searching for a specific type of resource. Different resource types have metadata and query techniques which are specific to that resource type. These resource-specific characteristics call for Discovery capabilities which capture those constraints.

The SIF subsets the Discovery Capability into four Capabilities; Resource Discovery, Content Discovery, Process Discovery, and Performer Discovery. This taxonomy of Discovery Capabilities is illustrated in Figure 14.

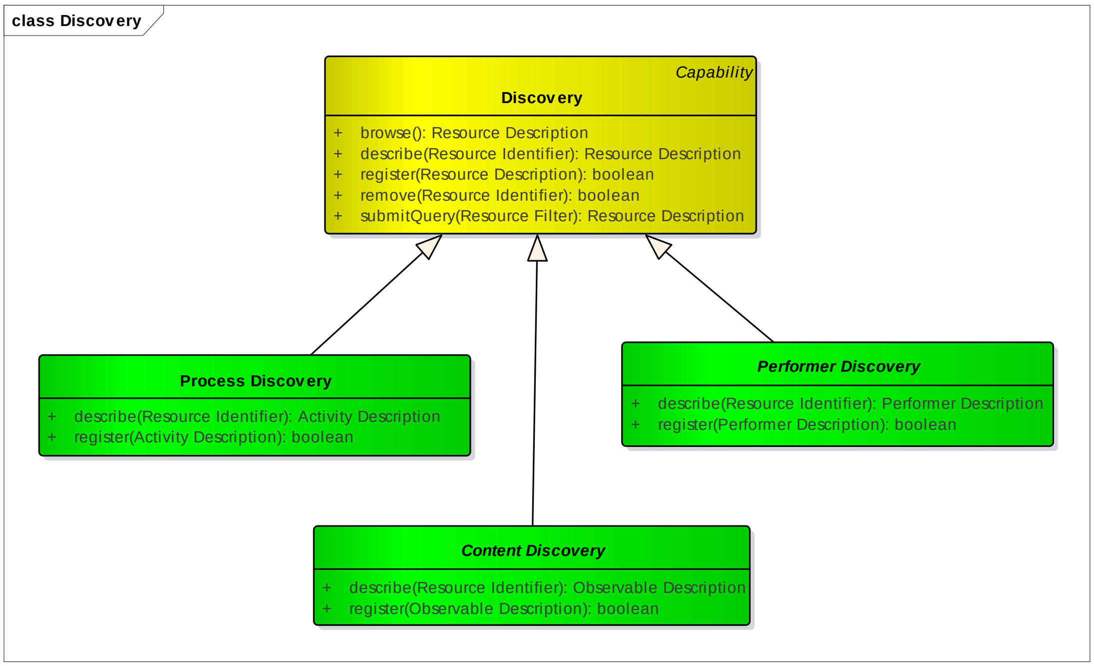


Figure 14 : Discovery Capabilities

9.1.2.1 Resource Discovery

Resource discovery is the ability to discover resources regardless of their type. These capabilities typically use a single metadata vocabulary to describe all resources. While that metadata may include the resource type, there are few if any elements which are specific to that type. Resource Discovery is useful to discovery the breath of the resources available, but it is not sufficient to narrow down the list to a few specific resources.

9.1.2.2 Content Discovery

Content discovery is the ability to discover information resources. These include data sets, files, and entries in relational databases. The Content Discovery metadata differs from Resource Discovery metadata in that it supports metadata elements and query terms which are specific to Information Resources. This allows the query to be better tuned for selection criteria useful to a user who is searching for content. However, an even greater difference is that the line between data and metadata can be ambiguous. Information Resources are designed to be read and processed. Much of the content of these resources can, and often is, used in query processing. Content Discovery capabilities support not just query processing against metadata. They can also process queries against the content itself.

Information Resources may be associated with specific Processes which are designed to process this information type.

9.1.2.3 Process Discovery

Process discovery is the ability to discover activities (See Section 9.1.7) which are available for execution. Processes encountered in sensor systems are typically software algorithms. Unlike Content the internals of a process cannot be queried directly. Therefore, there must be a metadata record to describe each process. This metadata includes three terms which are unique to process descriptions:

1) The inputs (including data types)
2) The outputs (including data types)
3) The controls or processing parameters (including data types)

A robust Process Discovery capability will also support discovery of complex processes. These are processes constructed from other processes. The complex process describes the order in which the sub-processes should execute and describes the connections of inputs to outputs.

Processes may be associated with specific Performers which are capable of executing that process.

9.1.2.4 Performer Discovery

Performers are the physical entities which execute all the Capabilities. They include servers, sensors, systems, people, and organizations. Performer Discovery is the ability to search for and receive descriptions of the performers available to a user.

A robust Performer Discovery capability will support discovery of complex systems. These are systems where the system itself as well as major sub-systems are independently discoverable.

Performers may be associated with specific Processes which are available to run on that Performer.

9.1.3 Delivery

Delivery is a category of capabilities which allows users to request and receive information resources which have been identified through Discovery. Delivery can be performed in three ways as illustrated in Figure 15.

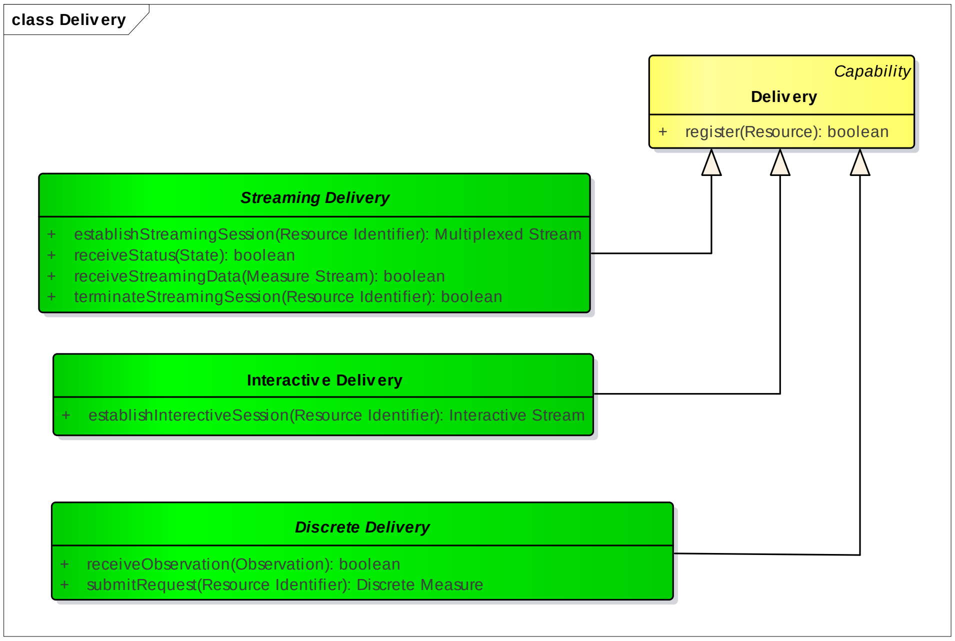


Figure 15 : Delivery Capabilities

9.1.3.1 Discrete Delivery

Discrete delivery is the delivery of an independent Information Resource. These resources can be delivered in two ways. The most familiar was is to deliver the resource itself. Reading the temperature value from a thermometer is this type of delivery. The second is to deliver a reference to the resource. When delivering a multi-gigabyte image, you might want to start with a URL to that image. The user can then access the resource when it is most convenient (i.e. fewer users on the network). The image is still a discrete delivery, just an indirect one.

9.1.3.2 Interactive Delivery

Interactive delivery allows the user to select which parts and how much of those parts to deliver at a time. At this time the JPEG 2000 Interactive Protocol (JPIP) is the only interactive delivery capability identified in the SIF.

9.1.3.3 Streaming Delivery

With streaming delivery, the client is provided a service access point from which the resource is available as a continuous stream of data. This form of delivery includes many Full Motion Video feeds, some audio feeds, and any real-time data feeds. The key discriminator to this method is that the data is a continuous stream.

9.1.3.4 Register

Submit a resource to a delivery service so that it will be available for access.

9.1.4 Command

Commands are used to start, modify, and stop the execution of processes. It is not clear that there is a valid scenario where a sensor owner will allow a remote user to command their sensor. Until a valid use case is identified, it would be premature to address commands in the SIF.

9.1.5 Sensing

The most important capability category for a Sensor Integration Framework would be the one having to do with Sensing. A simple concept which is deceptively complex.

In its simplest form, a sensor is a linear transfer function (an Activity or Process) which converts an environmental or physical phenomenon into an information resource. A thermocouple converts the temperature of an object (input) into a voltage (output). A thermocouple paired with an analog to digital converter is a sensing system which converts temperature (input) into a digital reading (output). This is a simple form of a passive, in-situ sensor system.

Most sensor systems can be classified based on two properties:

- In-situ vs. remote
- Active vs. passive

In-situ sensors are in or in close contact with the phenomenon they are measuring.

Remote sensors are at a distance from the phenomenon they are measuring.

Active sensors (RADAR, LIDAR) measure the reflection or propagation of an emitted signal. Note that the emitter does not have to be on the same platform as the detector.

Passive sensors (camera) measure the reflection or propagation of a naturally occurring signal.

This leads us to four possible classes of sensor:

1) In-situ passive
2) In-situ active
3) Remote passive
4) Remote active

Of these two, only the in-situ passive class includes sensors which do not require significant on-board processing. The Sensing capabilities must not only address emitters and detectors, it must also address the processing required to product usable information resources.

And then there are actuators. An actuator is the inverse of a sensor. It converts an Information Resource into an act upon the environmental. An automatic door, for example, will receive an information resource indicating that a person is present. That information is transformed into a rotation which moves a lever arm which opens the door. Other actuator types include the stepper motor used to rotate a sensor gimbal and a Harm missile.

In spite of this complexity, the Sensing capabilities fall into three classes; Command and Control, Sensor Collection, and Actuation. These three classes are illustrated in Figure 16.

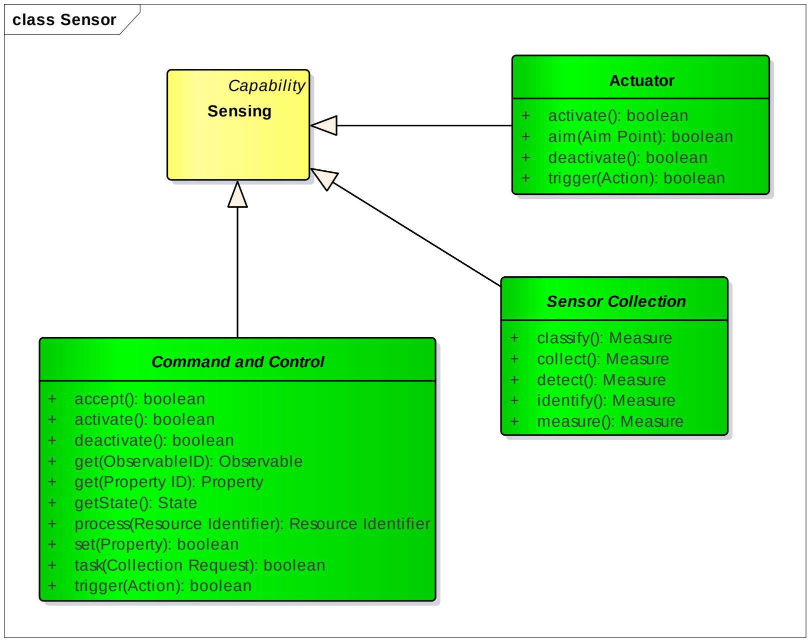


Figure 16 : Sensor Capabilities

9.1.5.1 Command and Control

Covers all of the control and status activities needed to place the sensor where it needs to be, establish the parameters for a successful collection, start and stop the collection.

9.1.5.2 Sensor Collection

Covers the activities performed by emitters, collectors, and all of the support processing required to produce a usable information resource.

9.1.5.3 Actuation

Currently a place holder. Will further populate in version 1.1

9.1.6 Human-Computer Interface

The Human-Computer Interface (HCI) category serves to capture the capabilities that should be available to a human user so that they may interact with the distributed sensing environment. These can best be visualized as a set of applets available on a user's desktop. These applets may:

Prepare information for input to the sensing environment

Present output from the sensing environment for Human consumption

Initiate an Activity within the sensing environment.

The HCI category is illustrated in Figure 17.

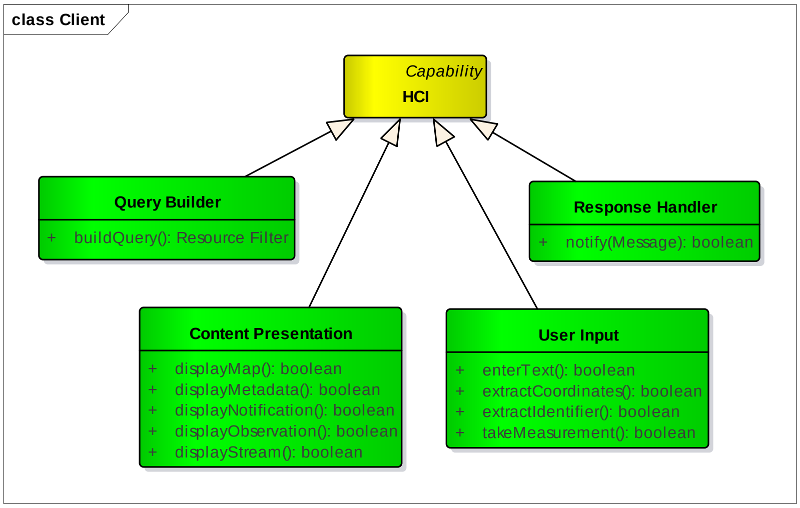


Figure 17 : HCI Capabilities

9.1.6.1 Query Builder

Discovery and Publish-Subscribe capabilities rely on a query expression to convey the rules for selecting results. Query expressions can be very complex. Query Builders are desirable as a means to simplify this process.

9.1.6.2 Content Presentation

The language of computers is ones and zeros. Even text is encoded in ones and zeros. Yet very few people can read binary. A binary dump of your favorite television program would not be very enjoyable. Tools are required to transform the raw data into a form which a human can comprehend. The Content Presentation capability addresses that need.

9.1.6.3 User Input

The language of computers is ones and zeros. Very few people can write in binary. User Input tools, such as a word processor, provide users the ability to enter information into a system using a method they are comfortable with. This capability also includes tools to extract information from one resource for use as input to another activity.

For example, drawing a bounding box on a map brings together capabilities. First of all, the map must be displayed using a Content Presentation capability. Then the user extracts the corner coordinates from the map using a User Input capability. Finally, the coordinates are input to a Query Builder to be added to a query expression.

9.1.6.4 Response Handler

Publish-subscribe and MoM messaging does not require that the client wait for a response. In fact, the client may receive a message they were not expecting. However, a message cannot be delivered unless there is some software entity there to receive it. Response Handlers are the capabilities for receiving and processing these messages.

9.1.7 Information Assurance

9.1.7.1 Identification

The process of discovering the true identity (i.e., origin, initial history) of a person or item from the entire collection of similar persons or items. In the SIF we discover the identity of Entities which can be a person (person entity) or a system (non-person entity).

9.1.7.2 Authentication

The act of verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system. Identification and Authentication are often treated as a single control, Identification and Authentication (I&A).

9.1.7.3 Authorization

The process of defining and maintaining the allowed actions. In an Attribute-Based Access Control (ABAC) environment, which is common in the IC, Authorization is the set of access privileges allocated to that entity. Having an access privilege does not mean that the entity will have access to a specific resource.

9.1.7.4 Access Control

Activities which permit or restrict access to applications at a granular level, such as per-user, per-group, and per-resources.

9.1.7.5 Confidentiality

Activities that protect data so that unauthorized parties cannot view the data.

9.1.7.6 Integrity

Activities which assure that data has not been altered in an unauthorized manner. Data integrity covers data in storage, during processing, and while in transit.

9.1.7.7 Non-Repudiation

Activities that provide protection against an individual falsely denying having performed a particular action. Provides the capability to determine whether a given individual took a particular action such as creating information, sending a message, approving information, and receiving a message.

9.2 Activities

DoDAF defines an Activity as “An action performed in conducting the business of an enterprise.” Perhaps more useful is its position in the DoDAF architecture:

An Activity is part of a Capability (the only parts in fact)

An Activity consumes a Resource (input)

An Activity produces a Resource (output)

An Activity performs under Conditions (parameters)

An activity is performed by a Performer

An Activity is constrained by Rules (standards and specifications)

While a Capability is the ability to do something, the Activities are the processes and process steps which make it happen. Within the SIF we assume that all Activities are performed by software. This is likely a valid assumption for sensing systems.

SIF also recognizes two forms of Activity. Atomic Activities are Activities which follow a request-response pattern. Processing Activities are Activities which will execute for a period (in some cases unbounded) of time. The SIF classifies Atomic Activities as Commands. Commands are described separately from the Processing Activities. Any discussion of Activities within the SIF will be referring to Processing Activities and not to Atomic Activities.

Since Capabilities are composed of Activities, then the logical way to organize Activities is by the Capabilities they implement.

Element	#	Type	Comments
Identifier	1..1	Resource Identifier	Unique identifier for this Activity
inputs	0..n	Input	Data to be input into the Activity
outputs	0..n	Output	Results to be received from the Activity
parameters	0..n	Control	Data to be used to control the execution of the Activity

Figure 18 : Activity Class Attributes

9.2.1 Discovery Activities

The activities shared by all Discovery Capabilities are described below.

9.2.1.1 Browse

The act of discovering new resources by following associations between the metadata cards.

Pre-Conditions	Possession of a metadata card or a reference to one.
Inputs	A reference to a metadata card
Outputs	A metadata card

9.2.1.2 Describe

The act of drilling down from a metadata card into more detailed information about a specific resource.

Pre-Conditions	Possession of an identifier for a resource.
Inputs	Resource identifier
Outputs	Resource metadata specific for that resource type

9.2.1.3 Register

The act of creating metadata for a resource and submitting that metadata to a Discovery Service.

Pre-Conditions	Possession of a metadata card or resource to register.
Inputs	The metadata card or resource to register
Outputs	Confirmation of registration

9.2.1.4 Submit Query

The act of submitting selection criteria to a Discovery Service and receiving metadata cards in response.

Pre-Conditions	A query expression has been created.
Inputs	The query expression
Outputs	Resource metadata or a metadata card

9.2.1.5 Remove

The act of removing a selected resource from the Discovery Service.

Pre-Conditions	A Resource has been identified and selected for deletion.
Inputs	The resource identifier
Outputs	Success or Failure

9.2.2 Discrete Delivery Activities

Discrete Delivery is the capability for a user to request an information resource and then receive a copy of that resource in a single transaction. An example of this is the use of HTTP Get to retrieve a web page.

9.2.2.1 Submit Request

The act of submitting a request for a resource or resources and receiving a representation of those resources in response.

Pre-Conditions	Client has an identifier for the desired resource
Inputs	The identifier for the desired resource
Outputs	The resource or an access path to the resource
Comments	In the case of a large resource it may be preferable to work with an access path rather than the entire resource.

9.2.3 Interactive Delivery Activities

Interactive Delivery is the capability to provide an information resource incrementally. The client is able to select the portions of the resource they want deliverer, specify when it should be delivered, and control the resolution of the delivered content. JPIP is an example of this capability.

9.2.3.1 Establish Interactive Session

The act of interactively selecting and retrieving portions of the identified resource.

Pre-Conditions	Client has an identifier for the desired resource.
Inputs	The identifier for the desired resource.
Outputs	The session identifier for the interactive session.
Post-Conditions	The session identifier is passed to a display client for the interactive protocol.

9.2.4 Streaming Delivery Activities

Streaming delivery is the capability to deliver an information resource incrementally in a time-synchronized manner. Motion Imagery is typically delivered in this way.

9.2.4.1 Establish Streaming Session

The act of establishing a connection to a data feed which will stream the resource to the user as a stream of content.

Pre-Conditions	Client has an identifier for the desired resource.
Inputs	The identifier for the desired resource.
Outputs	The session identifier for the streaming session.
Post-Conditions	The session identifier is passed to a display client for the steaming protocol.

9.2.5 Sensor Collection Activities

The Sensor Collection capability addresses the various ways that a sensor can collect and process data. Sensors can range from simple measurement devices to complex processing systems. The activities for this capability represent different ways a sensor collection capability can be implemented.

Note: The output from a sensor collection activity must be associated with metadata which describes the time and conditions under which the collection took place. In most cases that association is performed as a part of the collection activity.

9.2.5.1 Classify

The act of collecting an observation and comparing that observation against a set of patterns or templates with known entity types. If a match is observed, then the type of the observed individual or item is reported.

Inputs	A raw detection.
Processing	The raw detection is evaluated against a taxonomy of entity types.
Outputs	A value from the classification taxonomy.

9.2.5.2 Collect

The act of collecting multiple measurements which will be processed into the reported observation. Examples include Remote Imagery, LIDAR, and SIGINT.

Inputs	An input signal containing discrete measurements over time.
	Detailed telemetry from the sensor and platform time-synchronized with the signal.
Parameters	A sensor model and associated transformation parameters to convert from signal space to product space.
Processing	The raw signal is processed into a 2D or 3D representation of the measured values in an image or geospatial coordinate space.
Outputs	A coverage resource such as an image or a point cloud.

9.2.5.3 Detect

The act of determining that a pre-defined criterion has been met and issuing an alert that this is the case. This criterion is typically a threshold such as a maximum or minimum temperature. Detections can be sent directly to a sensor management system or distributed through a publish-subscribe capability.

Inputs	A continuous series of raw measurements.
Processing	Each raw measurement is evaluated against one or more criteria. If the criteria are met, then a detection is generated.
Outputs	A detection resource.

9.2.5.4 Identify

The act of collecting an observation and comparing that observation against a set of patterns or templates with known identities. If a match is observed, then the identity of the observed individual or item is reported. Biometrics are one form of Identifying activity.

Inputs	A raw detection.
Processing	The raw detection is evaluated against a taxonomy of identified entities.
Outputs	An identification resource for the observed target.

9.2.5.5 Measure

The act of measuring a physical quantity such as temperature.

Inputs	A raw detection.
Processing	The raw detection is converted into a digital representation.
Outputs	A measured value.

9.2.6 Sensor Command and Control Activities

The Sensor Command and Control capability provides means for a user to control the properties and activities of a sensor and sensor system.

9.2.6.1 Activate

The act of issuing a command to start a sensor Measure, Identify, Detect, or Classify activity.

Pre-Conditions	Client knows of an activity they wish to activate and the component where they want the activity to execute.
Inputs	- Component identifier - Activity identifier
Outputs	- Success or error indicator - Activity instance identifier

9.2.6.2 Deactivate

The act of issuing a command to stop a sensor Measure, Identify, Detect, or Classify activity.

Pre-Conditions	Client knows of an activity they wish to deactivate and the component where the activity is executing.
Inputs	- Component identifier - Activity instance identifier
Outputs	Success or error indicator

9.2.6.3 Get Observable

A command to retrieve an Observable from the sensor system.

Pre-Conditions	Client knows of an Observable they want to retrieve from a known component.
Inputs	- Component identifier - Observable identifier (may be a list of identifiers)
Outputs	- Success or error indicator - Observable (zero to many)

9.2.6.4 Get Property

A command to retrieve a Property from the sensor system.

Pre-Conditions	Client knows of a component and property they want to get the value of.
Inputs	- Component Identifier - Property name (may be a list of names)
Outputs	- Success or error indicator - Property name - Property Value (name and value may be a list of name-value pairs)

9.2.6.5 Get State

A command to retrieve the current State of the sensor system.

Pre-Conditions	Client knows of a component they want to get the status of
Inputs	Component identifier
Outputs	Component status metadata

9.2.6.6 Process

Many sensor systems include considerable processing capabilities. The act of issuing a process command takes advantage of those capabilities by instructing the sensor system to execute on-board processing of the detection data prior to delivery.

Pre-Conditions	Client has identified a process they want to run and a component where they want it to execute.
Inputs	- Component identifier - Process identifier - Process specific input data - Process specific control parameters
Outputs	- Success or error indicator - Process specific output data

9.2.6.7 Set Property

A command to set or modify a Property of the sensor system.

Pre-Conditions	Client wants to set a new value on a property.
Inputs	- Component Identifier - Property name - Property Value
Outputs	Success or error indicator

9.2.6.8 Task

The act of issuing a request to start a sensor Collect activity. This activity differs from triggering a collection in that the command is not sent to the sensor. Rather it goes to a sensor management system which may accept or reject the tasking request.

Pre-Conditions	Client has created a tasking request
Inputs	- An identifier for the component. - The tasking request.
Outputs	- Task acceptance or rejection status. - If accepted, return an identifier for the job.

9.2.6.9 Trigger

The act of issuing a command to execute a single Collection or Actuator activity.

Pre-Conditions	Client knows of an activity which can be triggered
Inputs	- Identifier for the component. - Identifier for the activity to be triggered. - May include activity-specific processing parameters.
Outputs	Success or error indicator

9.2.7 Actuation Activities

The Actuator category includes Capabilities which describe actions a system can take to impact its' environment. These Capabilities and Activities will be defined in a latter version.

9.2.8 Direct Messaging Activities

Direct messaging is the Capability for a user to send a message to a specified recipient or group of recipients.

9.2.8.1 Post Message

The act of creating the message, associating recipients with the message, and submitting the message for delivery.

Pre-Conditions	Client has a message to send and an intended recipient.
Inputs	The message and an identifier for the recipient.
Outputs	Success or error indicator.

9.2.8.2 Deliver Message

The act of delivering a message posted by an actor to the designated recipient.

Pre-Conditions	A message has been submitted for delivery
Inputs	The message and an identifier for the recipient.
Outputs	The message

9.2.9 Publish-Subscribe Messaging Activities

Publish-Subscribe Messaging is the Capability for a user to broadcast a message which can be received by anyone who may have an interest. Note that the Publish and Subscribe activities are instances of Direct Messaging being leveraged to support Publish Describe.

9.2.9.1 Publish

The act of creating a message, creating the associated metadata, and posting this content to the Publish-Subscribe service.

Pre-Conditions	Client has a message to share.
Inputs	The message and any required supporting metadata.
Outputs	Success or error indicator.

9.2.9.2 Subscribe

The act of creating a set of selection criteria and submitting that criteria to the Publish-Subscribe service.

Pre-Conditions	Client has created a query expression for selecting messages to receive.
Inputs	- The query expression - The identifier for the response handler to receive notifications
Outputs	- Success or error indicator. - A subscription identifier

9.2.9.3 Unsubscribe

The act of removing selection criteria from a Publish-Subscribe service thereby canceling receipt of messages which match that criteria.

Pre-Conditions	Client has the subscription identifier for a subscription they would like to cancel.
Inputs	The subscription identifier
Outputs	Success or error indicator.

9.2.9.4 Notify

When a message meets the selection criteria of a subscriber, the Publish-Subscribe service will perform the act of Notifying the subscriber of the existence of the message and provide a means for the subscriber to access the message.

Pre-Conditions	Publish-subscribe service has active subscriptions.
Inputs	A publish request.
Outputs	The published message routed to all response handlers associated with subscriptions which meet the selection criteria.

9.2.10 Message Oriented Middleware Activities

Message Oriented Middleware is a Messaging Capability where the routing decisions are made by the messaging infrastructure, not the sender or recipient. Routing of messages are typically determined by a rule set.

9.2.10.1 PostMessage

The act of creating the message and submitting the message for delivery. This activity differs from that of direct messaging in that there is no need to associate recipients with the message. That function is performed by the Messaging Capability.

Pre-Conditions	Client has a message to send.
Inputs	The message and any required supporting metadata.
Outputs	Success or error indicator.

9.2.10.2 RouteMessage

Route Message is an internal activity of the messaging service. It captures the responsibility for the messaging service to determine appropriate recipients for all posted messages.

Pre-Conditions	The MoM service has received a message and possesses a routing rule set.
Inputs	The message and any required supporting metadata.
Outputs	The message directed to recipients in accordance with the rule set.

9.2.10.3 Deliver

Provide the message to the selected recipients.

Pre-Conditions	Client has registered a response handler with the MoM system.
Inputs	A message to deliver.
Outputs	The message delivered to the Response Handler.

9.2.11 Information Assurance Activities

This has been deferred until a later version of the SIF-SP.

9.2.12 Human Computer Interface Activities

9.2.12.1 Query Builder

The Query Builder capability assists users in building Resource Filters.

9.2.12.1.1 Build Query

The act of building a Resource Filter based on user supplied criteria.

Pre-Conditions	None.
Inputs	User selected query components, often generated through a set of pull-down menus.
Outputs	A query expression
Options	User Input activities can be used to add spatial-temporal and derived data.

9.2.12.2 Response Handler

The Response Handler capability functions as a listener for Alerts, Notifications, etc.

9.2.12.2.1 Notify

Notify is the activity of receiving a message from a notifying source.

Pre-Conditions	Client has registered a response handler with a messaging system.
Inputs	The message to deliver.
Outputs	An indicator that a message has been received.

9.2.12.3 Content Presentation

The Content Presentation capability provides a means to display content for use by a user.

9.2.12.3.1 Display Map

The act of displaying the content on a map. This typically involves indicating the location of the observation through a marker on a map display. The marker is visually coincident with the location of the observed event or the sensor which made the observation.

Pre-Conditions	Client has received a coverage resource.
Inputs	The coverage resource.
Outputs	A graphic display of the coverage.

9.2.12.3.2 Display Metadata

The act of displaying metadata resulting from a Discovery or Describe activity.

Pre-Conditions	Client has registered a response handler with the MoM system.
Inputs	A message to deliver.
Outputs	The message delivered to the Response Handler.
Options	Response handler may display the message or raise an indicator allowing the user to display the message at a later time.

9.2.12.3.3 Display Notification

The act of displaying a notification from a publish-subscribe or direct messaging activity.

Pre-Conditions	Response handler has received a messaging and the user has elected to view it.
Inputs	The message to view.
Outputs	A text representation of the message.

9.2.12.3.4 Display Observation

The act of displaying an Observation or Measure.

Pre-Conditions	Client has received an observation.
Inputs	The observation
Outputs	A display of the observation content.
Options	Observations may be displayed geospatially using the Display Map Activity.

9.2.12.3.5 Display Interactive Stream

The act of displaying an interactive stream.

Pre-Conditions	Client has discovered an interactive delivery end-point.
Inputs	- An interactive delivery end-point with information to provide. - Client generated commands to control the information stream
Outputs	A visual representation of the streamed resource.

9.2.12.3.6 Display Stream

The act of displaying streaming data.

Pre-Conditions	Client has discovered a streaming delivery end-point.
Inputs	A streaming delivery end-point with information to provide.
Outputs	A visual representation of the streamed resource.

9.2.12.4 User Input

The User Input capability provides means for a user to input data into the system.

9.2.12.4.1 Extract Coordinates

The act of extracting coordinates from an information source and using them in a derivative product. The bounding box used in a query expression could be extracted from the map display.

Pre-Conditions	Client has received an image or similar resource which can be displayed in a spatially correct manner.
Dependencies	Requires content presentation to display measured resource

Inputs	- An image or similar spatially rectified resource - Manual selection of points using a mouse or similar device
Outputs	A spatial-temporal resource

9.2.12.4.2 Take Measurement

The act of extracting spatial information from an information source. Measuring the length of a building displayed on an Electric Light Table is an example of a Take Measurement activity.

Pre-Conditions	Client has received an image or similar resource which can be displayed in a spatially correct manner.
Dependencies	Requires content presentation to display measured resource
Inputs	- An image or similar spatially rectified resource - Manual selection of points using a mouse or similar device
Outputs	A measurement resource

9.2.12.4.3 Enter Text

The act of entering text using a keyboard.

Inputs	Manual input via a keyboard or similar device.
Outputs	A text data resource.

10 Engineering and Technology Viewpoints

The Engineering and Technology Viewpoints define the technical constraints on an architecture. However, these views are not defined in a vacuum. Few modern systems are "clean sheet" developments. They all start with a robust technical environment within which they must operate. These environments can be identified and scoped based on their known properties. They are well enough defined that it is possible to build a taxonomy of technical environments.

This Sensor Integration Framework Standards Profile builds such a taxonomy for sensors and sensor processing systems. Each element in the taxonomy represents a unique set of enabling technologies and constraints. These in turn restrict the set of possible design solutions which can be developed for that environment. Such a taxonomy also serves the need to identify and scope the Technical Views of the Profile. To provide interoperability across defined technical views, the enterprise technical view serves as the common target for data sharing and all other technical views include definitions for bridging information from that view into the enterprise view.

The primary constraint on sensor integration is the communications environment within which the sensor has been deployed. An analysis of the distinguishing properties of communications environments is provided in Annex C. That analysis identifies the following distinguishing properties:

- Secure: The parties can engage in trusted, private information exchanges
- Disconnected: Network connectivity may be lost.
- Low Bandwidth: Network throughput may be very low.
- Segmented: Only a portion of the network may be available. It is no longer global.
- Unreliable: A large percentage of the messages may be lost or corrupted.
- Mobile: At least one party is moving. Network properties will change as the network adapts to the changing locations.

Other properties; such as size, weight, and power (SWAP) are also important constraints. However, it is a tendency of system designers to use the most capable communications environment possible, while minimizing the impact that environment has on SWAP. As a result, the technical constraints of the communications environment has a larger impact than those for SWAP and other technical constraints.

The SIF-SP identifies five communications environments:

- Enterprise – IP with Web Services and global resources
- Enclave – IP with Web Services but isolated from global resources
- Tactical DDIL IP based networks
- CDL – Airborne non-IP networks
- IEEE 1451 – Wire protocols

These five form a rough progression from the most capable (Enterprise) to the least (1451). The following sections describe each environment.

10.1 Enterprise

This is the World Wide Web as it is commonly used.

- Secure: Multiple security regimes supported by globally accessible services
- Disconnected: Effectively always on
- Low Bandwidth: Effectively unlimited
- Segmented: No – global reach
- Unreliable: No – error detection and correction resolves most errors
- Mobile: Environment accommodates mobile nodes, making mobility invisible to the users

10.2 Enclave

Same basic technology as the Enterprise, but with a limited reach.

- Secure: Usually operates under a single homogeneous set of security services and policies
- Disconnected: Effectively always on
- Low Bandwidth: Effectively unlimited
- Segmented: Yes – access is limited to resources within the enclave. WWW resources may not be accessible.
- Unreliable: No – error detection and correction resolves most errors
- Mobile: Environment accommodates mobile nodes, making mobility invisible to the users

10.3 Tactical DDIL IP

This environment also employs the basic technology of the Enterprise or Enclave, but with significant limited capabilities and environmental constraints that preclude full use of selected technical approaches which require robust environments.

- Secure: Tactical domain necessitates authentication of communicating components, authorization of requested actions and confidentiality of in-transit communications.
- Disconnected: Subject to interruption of communication and inability to establish connectivity, requiring an ability to operate properly when disconnected from other nodes.
- Low Bandwidth: Significantly constrained communication throughput, necessitating use of bandwidth-efficient data encoding and limitations on the size and frequency of data transmission.
- Segmented: Both the entire environments and subsets of sensors and nodes will be limited in ability to maintain access within the environment and with the enterprise.
- Unreliable: Packet loss and corruption is common, requiring use of reliable communication mechanisms.
- Mobile: Tactical environment requires support to both static and mobile nodes. Mobility may impact and exacerbate security, bandwidth, connectivity, and reliability.

10.4 CDL

Common Data Link (CDL) is a secure protocol commonly used by airborne assets to communicate with each other and with ground stations. CDL links have the following properties:

- Secure: Encrypted with anti-jamming and anti-intercept
- Disconnected: Line-of-sight. Connection is lost if both nodes are not visible to each other.
- Bandwidth: Megabit +
- Segmented: Point-to-point.
- Unreliable: Error detection and correction resolves most errors
- Mobile: Nodes are usually mobile, but this mobility is accommodated by the environment.

10.5 IEEE 1451

Wire protocol connecting very small sensors to a node. This node often serves as a Point of Presence (POP) in a more capable environment.

- Secure: Not usually provided. Encryption is sometimes used.
- Disconnected: May be.
- Low Bandwidth: May be in the low Kilobits
- Segmented: Yes. Scope is limited to the sensors connected to a node.
- Unreliable: Yes. Errors are seldom detected and rarely corrected.
- Mobile: No – sensors are hard-wired to the node

11 Environment Integration

The goal of the SIF-SP is to enable enterprise integration of sensors and sensor systems. It is not enough to identify the standards and guidance governing sensors within a communications environment, the SIF-SP must also specify how to integrate across communications environments.

The communications environments can be organized into a hierarchy. Those at the top of the hierarchy have greater capabilities than those below. In practice, any environment can inter-connect to any other. However, it is sufficient to specify how a low capability environment interconnects with the next more capable environment.

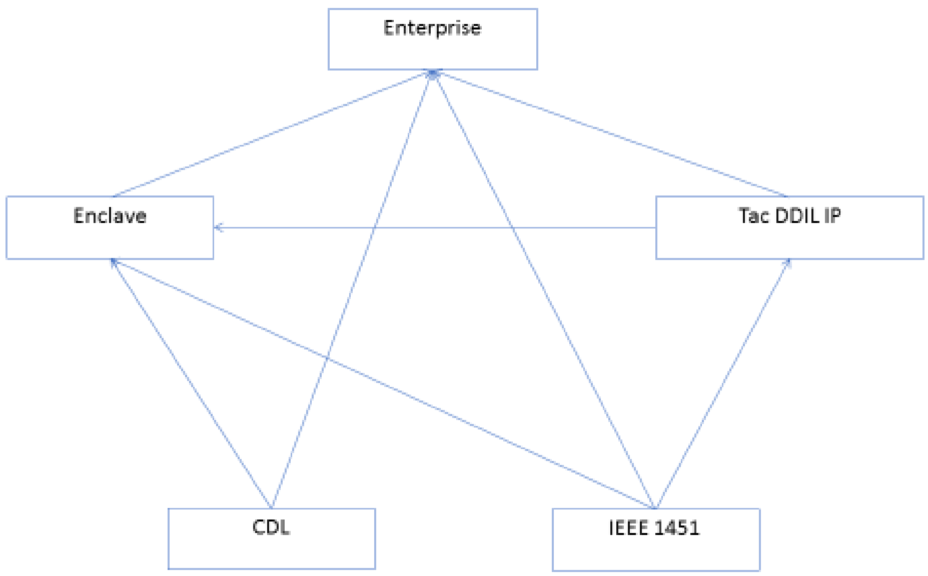


Figure 19 : SIF-SP Environments

To provide interoperability across defined technical views, the enterprise technical view serves as the common target for data sharing and all other technical views include definitions for bridging information from that view into the enterprise view. Each of the other technical views shall define conformance classes for basic capabilities within the specified communications environment and for bridging to provide interoperability with the enterprise technical view. Compliance with the bridging conformance class is conditionally required when interconnecting beyond the specified communications environment to exchange information in the enterprise communication environment

Annex A Terms and Definitions

For the purpose of this document, the following terms and definitions apply:

Actuator <SensorML 2.0>
A type of transducer that converts a signal to some real-world action or phenomenon.

Bounding Box <OWS Common 2.0>
The portion of a coordinate space that lies between a lower bound and an upper bound in each dimension of a coordinate reference system.

NOTE A bounding box can be used to express spatial-temporal query constraints, or to describe the (approximate) location and extent of geospatial data. A bounding box is often called the "minimum bounding rectangle" of a geospatial data item when its lower and upper bounds in each dimension are those of the data item.

EXAMPLES: Rectangle in two spatial dimensions, rectangular solid in three spatial dimensions.

Client <OWS Common 2.0>
Software component that can invoke an operation from a server.

Consumer
Synonym: event sink <SWE Service Model>
Receiver of notifications sent from other components; role in the event architecture. (see [OGC 09-032]).

Coordinate Reference System (CRS) <SensorML 2.0>
A spatial or temporal framework within a position and/or time can be defined. According to ISO 19111, a coordinate system that is related to the real world by a datum.

Data Component <SensorML 2.0>
Element of sensor data definition corresponding to an atomic or aggregate data type

Note: A data component is a part of the overall dataset definition. The dataset structure can then be seen as a hierarchical tree of data components.

Datum <SensorML 2.0>
Undefined in ISO 19111. Defined here as a means of relating a coordinate system to the real world by specifying the physical location of the coordinate system and the orientation of the axes relative to the physical object. For a geodetic datum, the definition also includes a reference ellipsoid that approximates the physical or gravitational surface of the planetary body.

Detector <SensorML 2.0>
Atomic part of a composite Measurement System defining sampling and response characteristic of a simple detection device. A detector has only one input and one output, both being scalar quantities. More complex Sensors, such as a frame camera, which are composed of multiple detectors, can be described as a detector group or array using a System or Sensor model.

Event <SWE Service Model>

Anything that happens or is contemplated as happening at an instant or over an interval of time. [OGC 09-032].

Event Object <SWE Service Model>

Object that represents, encodes, or records an event, generally for the purpose of computer processing. [see OGC 09-032].

Feature <SensorML 2.0>

Abstraction of real-world phenomena [ISO 19101:2002, definition 4.11].

Note: A feature may occur as a type or an instance. Feature type or feature instance should be used when only one is meant.

Interface <OWS Common 2.0>

Named set of operations that characterize the behavior of an entity [ISO 19119].

Location <SensorML 2.0>

A point or extent in space relative to a coordinate system. For point-based systems, this is typical expressed as a set of n-dimensional coordinates within the coordinate system. For bodies, this is typically expressed by relating the translation of the origin of an object's local coordinate system with respect to the origin of an external reference coordinate system.

Measure (noun) <SensorML 2.0>

Value described using a numeric amount with a scale or using a scalar reference system [ISO/TS 19103]. When used as a noun, measure is a synonym for physical quantity.

Measurement (noun) <SensorML 2.0>

An observation whose result is a measure [O&M].

Measurement (verb) <SensorML 2.0>

An instance of a procedure to estimate the value of a natural phenomenon, typically involving an instrument or sensor. This is implemented as a dynamic feature type, which has a property containing the result of the measurement. The measurement feature also has a location, time, and reference to the method used to determine the value. A measurement feature effectively binds a value to a location and to a method or instrument.

Measurement <SOS 2.0>

Set of operations having the object of determining the valueof a quantity. [ISO/TS 19101-2:2008, definition 4.20].

Notification <SWE Service Model>

Synonym: message.
Container for event objects. [see OGC 09-032].

Observable, Observable Property (noun) <SensorML 2.0>

A parameter or a characteristic of a phenomenon subject to observation. Synonym for determinand.[O&M].

A physical property of a phenomenon that can be observed and measured (e.g. temperature, gravitational force, position, chemical concentration, orientation, number-of-individuals, physical switch status, etc.), or a characteristic of one or more feature types, the value for which will be estimated by application of some procedure in an observation. It is thus a physical stimulus that can be sensed by a detector or created by an actuator.

Observation <SensorML 2.0>
Act of observing a property or phenomenon [ISO/DIS 19156, definition 4.10].

Note: The goal of an observation may be to measure, estimate or otherwise determine the value of a property.

Observed Property <SOS 2.0>
Facet or attribute of an object referenced by a name [OGC 10-004r3/ISO 19156] which is observed by a procedure.

Operation <OWS Common 2.0>
Specification of a transformation or query that an object may be called to execute [ISO 19119].

Orientation <SensorML 2.0>
The rotational relationship of an object relative to an external coordinate system. Typically expressed by relating the rotation of an object's local coordinate axes relative to those axes of an external reference coordinate system.

Parameter <OWS Common 2.0>
Variable whose name and value are included in an operation request or response.

Phenomenon <SensorML 2.0>
A physical state that can be observed and its properties measured.

Platform <OWS Common 2.0>
The underlying infrastructure in a distributed system (Adapted from ISO 19119)

NOTE A platform describes the hardware and software components used in a distributed system. To achieve interoperability, an infrastructure that allows the components of a distributed system to interoperate is needed. This infrastructure, which may be provided by a Distributed Computing Platform (DCP), allows objects to interoperate across computer networks, hardware platforms, operating systems and programming languages. (Adapted from Subclause 10.1 of ISO 19119).

Position <SensorML 2.0>
The location and orientation of an object relative to an external coordinate system. For body-based systems (in lieu of point-based systems) is typically expressed by relating the object's local coordinate system to an external reference coordinate system. This definition is in contrast to some definitions (e.g. ISO 19107) which equate position to location.

Procedure <SOS 2.0>
Method, algorithm, instrument, sensor, or system of these which may be used in making an observation. [OGC 10-004r3/ISO 19156].

NOTE: As the definition of procedure states, this standard uses that term as a generalization of, for example, the terms sensor and sensor system, but also for simulations or other calculations that may produce observations.

Process <SensorML 2.0>
An operation that takes one or more inputs, and based on a set of parameters, and a methodology generates one or more outputs.

Producer <SWE Service Model>
Specialization of a publisher that offers an additional subscription interface; role in the event architecture. (see [OGC 09-032]).

Property <SensorML 2.0>
Facet or attribute of an object referenced by a name. [ISO/DIS 19143:2010].

Example : Abby's car has the color red, where "color" is a property of the car instance, and "red" is the value of that property.

Publisher <SWE Service Model>
Sends notifications to other components; role in the event architecture (see [OGC 09-001]).

Reference Frame <SensorML 2.0>
A coordinate system by which the position (location and orientation) of an object can be referenced.

Request <OWS Common 2.0>
Invocation of an operation by a client.

Response <OWS Common 2.0>
Result of an operation, returned from a server to a client.

Resource <OWS Common 2.0>
Any addressable unit of information or service. [IETF RFC 2396].

EXAMPLES: Files, Images, Documents, Programs, and Query Results.

NOTE The means used for addressing a resource is a URI (Uniform Resource Identifier) reference.

Result <SensorML 2.0>
An estimate of the value of some property generated by a known procedure. [O&M].

Sensor <SensorML 2.0>
An entity capable of observing a phenomenon and returning an observed value. Type of observation procedure that provides the estimated value of an observed property at its output.

Note: A sensor uses a combination of physical, chemical or biological means in order to estimate the underlying observed property. At the end of the measuring chain electronic devices often produce signals to be processed.

Sensor Model <SensorML 2.0>

In line with traditional definitions of the remote sensing community, a sensor model is a type of Location Model that allows one to georegister or co-register observations from a sensor (particularly remote sensors).

Sensor Data <SensorML 2.0>
List of digital values produced by a sensor that represents estimated values of one or more observed properties of one or more features.

Note: Sensor data is usually available in the form of data streams or computer files.

Sensor System <SOS 2.0>
System whose components are sensors. A sensor system as a whole may itself be referred to as a sensor with an own management and sensor output interface. In addition, the components of a sensor system are individually addressable. [OGC 06-021r4].

(Sensor) Platform <SensorML 2.0>
An entity to which can be attached sensors or other platforms. A platform has an associated local coordinate reference frame that can be referenced relative to an external coordinate reference frame and to which the reference frames of attached sensors and

Service <OWS Common 2.0>
Distinct part of the functionality that is provided by an entity through interfaces. [ISO 19119] .

capability which a service provider entity makes available to a service user entity at the interface between those entities. [ISO 19104 terms repository].

Subscription <SWE Service Model>
Represents the relationship between consumer and producer, including any content or channel filter, along with any relevant policy and context information. (compare with OASIS WS-BaseNotification).

Value <SensorML 2.0>
A member of the value-space of a datatype. A value may use one of a variety of scales including nominal, ordinal, ratio and interval, spatial and temporal. Primitive datatypes may be combined to form aggregate datatypes with aggregate values, including vectors, tensors and images. [ISO11404].

Version <OWS Common 2.0>
Version of an Implementation Specification (document) and XML Schemas to which the requested operation conforms.

NOTE: An OWS Implementation Specification version may specify XML Schemas against which an XML encoded operation request or response must conform and should be validated.

Annex B **Abbreviations**

In this document the following abbreviations and acronyms are used or introduced:

ABAC	Attribute Based Access Control
CBRN	Chemical, Biological, Radiological, and Nuclear
CDL	Common Data Link
CIO	Chief Information Officer
CRS	Coordinate Reference System
DDIL	Denied, degraded, intermittent, or limited bandwidth
DM2	DoDAF Metadata Framework
DoD	Department of Defense
DoDAF	DOD Architecture Framework
FG	Focus Group
FMV	Full Motion Video
GWG	Geospatial-Intelligence Standards Working Group
GWS	Geospatial Web Service
HCI	Human Computer Interface
IC	Intelligence Community
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISA	Integrated Sensor Architecture
ISO	International Organization for Standardization
ITU-T	Telecommunication Standardization Sector of the International Telecommunications Union
JESC	Joint Enterprise Standards Committee
JPIP	JPEG 2000 Interactive Protocol
LIDAR	Light Detection and Ranging
MODAF	Ministry of Defence Architecture Framework (UK)
MoM	Message oriented middleware
NGA	National Geospatial-Intelligence Agency
NSG	National System for Geospatial-Intelligence
O&M	Observations and Measurements

OASIS-WS	Organization for the Advancement of Structured Information Standards – Web Security
ODNI	Office of the Director of National Intelligence
OGC	Open Geospatial Consortium
ODP	Open Distributed Processing
ORM	OGC Reference Model
OWL	Web Ontology Language
OWS	OGC Web Services
POP	Point of Presence
RM-ODP	Reference Model of Open Distributed Processing
RPC	Remote Procedure Call
SensorML	Sensor Model Language
SIF	Sensor Integration Framework
SIGINT	Signals Intelligence
SIF-SP	Sensor Integration Profile
SOS	Sensor Observation Service
SSN	Semantic Sensor Network
SWAP	Size, Weight, and Power
SWE	Sensor Web Enablement
UML	Unified Modeling Language
USMS	US MASINT System
W3C	World Wide Web Consortium
XML	eXtended Markup Language
2D	Two Dimensional
3D	Three Dimensional

Annex C Properties of Communications Environments

C.1 Background

An underlying assumption to Web technologies is the availability of a global, high bandwidth communications infrastructure such as the Internet. That assumption is not always valid. The tasks described in this section examine ways OGC Services can be adapted for Denied, Degraded, Intermittent, or Limited bandwidth (DDIL) communications environments.

The Internet provides a reliable, high speed, global information exchange service. Most of us take it for granted. Yet the Internet is a very complex system. It can and does fail often. Furthermore, it is not truly global. There are communities within 70 miles of Washington D.C. which have little or no access to the Internet. Therefore, the OGC Web Services standards should be extended to address communications environments with the following properties:

- Secure: The parties can engage in trusted, private information exchanges
- Disconnected: Network connectivity may be lost.
- Low Bandwidth: Network throughput may be very low.
- Segmented: Only a portion of the network may be available. It is no longer global.
- Unreliable: A large number of the messages may be lost or corrupted.
- Mobile: At least one party is moving. Network properties will change as the network adapts to the changing locations.

C.2 Secure

First and foremost, Web Services must be secure. They need to be able to identify the requesting party, authenticate that identity, maintain the confidentiality of the exchange, enforce access control for the hosted resources, and provide non-repudiation for the actions taken by a user. The technologies to provide these capabilities are mostly defined in non-OGC standards and technologies. Yet they must be part of a deployed OGC service and must be accommodated by OGC standards

C.3 Disconnected

Devices are not always connected to the network. Yet they are still expected to perform their function. Technical strategies can be developed to deal with disconnected operation. The nature of the strategy depends on the reason the device is disconnected and expected reconnection scenarios. These include:

- Network connectivity is not available but will be available at a later date
- Device disconnects to preserve power but will reconnect when “woken up”
- Mobile device which is temporally out of range of a Point of Presence (POP)
- Passive device which only connects when interrogated
- Burst device which only connects at certain times of day for a fixed period
- Intermittent device which due to poor line quality fades in and out

C.4 Low Bandwidth

Internet speeds in excess of 100 Mbps are common. Many Web services are designed for this bandwidth and higher. Yet not all devices have this bandwidth at their disposal. ZigBee, a common IoT protocol, runs at 250 Kbps. Some tactical military networks run at 9,600 bps.

Low bandwidth is often the result of limitations in Size Weight and Power (SWAP). These limitations also restrict the options for maximizing bandwidth utilization. Any solution which increases the computational load draws power, which goes against the limited available SWAP. Optimal solutions would maximize bandwidth utilization with little or no increase in SWAP.

C.5 Segmented

Web services assume the Internet is globally connected. That is not the case. Many organizations use guards and firewalls to control the information that flows in and out of their Intranet. Resources within the Intranet may not be accessible to users outside of it. Likewise, access to Internet resources may not be allowed from the Intranet. This form of segmentation can be anticipated and planned for.

Another form of segmentation is the result of unanticipated events. These can be equipment failure, a denial of service attack, or natural disaster. The result is a communications environment which works fine within a limited geographic or cyber community. But has no access to resources outside of that scope.

C.6 Unreliable

Information is exchanged over the Internet using fixed size packets. Great efforts are taken to make sure that all packets are delivered, without error, in the order they are expected. However, at its heart the Internet is just a signal flowing over a cable. Noise, interference, even active jamming can corrupt that signal. Internet protocols correct most of these errors, but messages still get lost.

The situation is even worse on DDIL networks. Lower bandwidth, a noisier and more hostile environment, and SWAP imitations preclude many of the error detection and recovery techniques used by the Internet. Communications errors are common. Participants on a DDIL network must be able to perform their own error detection and recovery.

C.7 Mobile

Internet protocols are very good at adapting to changes in network topology. That is what allows smart phones to stay connected even when traveling down the highway at 70 MPH. However, this adaptability is only made possible through a very complex infrastructure. An infrastructure which doesn't exist in many parts of the world.

The property of mobility has implications for the other properties as well. For example, Mobile Ad-Hoc Networks (MANETs) are continuously self-configuring, infrastructure-less networks of mobile devices connected using wireless radios. Since they use a peer-to-peer message exchange over a mesh network, there is no requirement for a complex infrastructure. All that is necessary is that one of the nodes on the mesh have access to an outside network. However, they can suffer from high incidents of disconnection, limited bandwidth, and high error rates. They are also by nature segmented networks, self-contained unless one node has a link to the outside world.

C.8 Network Topologies

The following set of network topologies provide some insight into the complexity involved in network communications. The Internet uses all of these in one place or another. A DDIL network is most likely going to be a star, point-to-point, or mesh network. When DDIL networks connect to the Internet, they usually use a star configuration. A single node which has connectivity to both the DDIL and Internet serves as a bridge between the two environments.

Bus: A network topology which consists of a shared backbone to which all nodes are connected. Communications between nodes are performed over the shared backbone. Since it is a shared resource, this topology requires a low-level protocol to manage access to the backbone.

Daisy Chain: A series of point-to-point connections.

Hierarchical: This network topology can be visualized as a tree of star networks. The nodes at each level manage the nodes at the next level down.

Mesh: Mesh topologies are commonly used for ad-hoc networks. There is no centralized delivery system. When a node wants to send a message, it sends that message to all of its nearest neighbors. The neighbors forward it to their neighbors, and so on until it gets (or doesn’t get) to the intended recipient. This structure makes it easy for nodes to enter and leave the network without disrupting the other nodes.

Point-to-Point: A point to point network consists of two nodes connected by a channel. All communications between the two nodes take place over that channel. Communication with any other node is not supported.

Ring: A ring network is a form of Bus network where the ends of the bus come together to form a ring. This form of network often uses a rotating token to manage access to the backbone.

Star: A network topology which consists of one central node supporting multiple satellite nodes. The satellite nodes have a point-to-point relationship with the central node. The central node is responsible for routing messages from the originator to the destination.